CURSO DE INTRODUCCIÓN A LA PROGRAMACIÓN SAS® v9

Versión de manual 2.5

Universidad Complutense de Madrid
Servicios Informático de Apoyo a Docencia e Investigación
Apoyo a Investigación
(Revisado noviembre 2009)

Introducción

Esta guía contiene un pequeño manual de SAS® V9 y supone una actualización de la versión de manual anterior, referente a SAS® V8

En general, esta guía está muy orientada a la sintaxis de programación de SAS®.

En SAS existen utilidades que pueden realizarse con la interfaz (entorno gráfico) de usuario y que se presentan en esta guía. De todas ellas, siempre que sea posible, indicaremos cómo generar el código SAS® correspondiente. La obtención del código SAS® permite almacenarlo en un fichero y poder ejecutarlo de nuevo en otra sesión SAS, sin necesidad esta vez de realizar de nuevo los pasos con el entorno gráfico.

Las utilidades de entorno gráfico que permite SAS son:

Creación y manipulación de librerías de datos SAS (que llamaremos SAS-FILES)

Importación/exportación de datos externos en ficheros ASCII

Importación/exportación de datos externos en ficheros en otros formatos, por ejemplo, EXCEL.

Exportación de gráficos a otros formatos

Esta versión del manual incorpora en el texto los apéndices de la versión anterior relativos al entorno gráfico de SAS.

Dada la extensión del documento en relación con la duración del curso de introducción que se imparte, algunas utilidades u opciones más avanzadas de versiones previas de este documento se han expresado mediante notas a píe de página o mediante la indicación de AVANZADO. Por este motivo, las notas a píe de página no avanzadas se señalan con negrita.

Cambios con la versión de manual v2.0: Incorpora la salida de gráficos en período de pruebas de algunos procedimientos estadísticos, a partir de programación ODS. Véase sección VI.III.

INTRODUCCIÓN AL PAQUETE ESTADÍSTICO SAS®

I. GENERALIDADES DEL SISTEMA SAS®

I.1 VISIÓN GENERAL DEL SISTEMA SAS®

El sistema SAS proporciona numerosas herramientas para el análisis estadístico de datos y la generación de informes.

Además del SAS BASE y del SAS/STAT, la Universidad Complutense de Madrid cuenta con licencia de los módulos SAS/GRAPH para la realización de gráficos, SAS/OR de investigación operativa, SAS/ETS para análisis de series temporales, SAS/IML para cálculos matriciales, SAS/FSP para construcción de paneles de entrada de datos, SAS/AF para desarrollo de aplicacines, SAS/QC para control de calidad, SAS/INSIGHT para inspección visual de datos, SAS PC File Formats y SAS/EIS (Executive Interactive System), SAS/CONNECT para ejecución de SAS en remoto, SAS MDDB Server common products.

La Universidad cuenta así mismo con la solución de SAS, Enterprise Miner[1] y con el producto Enterprise Guide, ambas con un entorno gráfico que no requiere conocimientos de la sintaxis de SAS.

La página url de SAS es *http:/www.sas.com/* y de SAS España *http:/www.sas.com/europe/spain*.

I.2 CLASIFICACIÓN DE LOS PROGRAMAS DE ESTADÍSTICA DE LOS MODULOS SAS/BASE® Y SAS/STAT®

SAS permite la ejecución de algunos procedimientos no sólo en el entorno interactivo, sino que además esta ejecución sea de hecho interactiva. Es decir, se pueden ejecutar algunos comandos del procedimiento y posteriormente otros comandos del mismo. (Algunos procedimientos son: ANOVA, CATMOD, GLM, PLAN, REG, etc…)

Veamos sucintamente los procedimientos disponibles para el análisis de datos en esta lista de procedimientos no exhaustiva. Procedimientos nuevos en la versión 9, se señalan con el símbolo 9.1.

Estadística descriptiva univariante.

MEANS	Descripción univariante.
UNIVARIATE	Descripción univariante. Diagramas de 'tallo-hoja' y de 'caja'. Percentiles, test de normalidad.
CORR	Matrices de correlación entre variables.
CHART	Gráficos sectoriales, en estrella, de barras y por bloques. Visualización gráfica de varias variables simultáneamente.
PLOT	Gráficos bidimensionales.

[1] **ATENCIÓN: Esta solución de SAS es temporal y condicionada a la docencia que se haga de ella. Los profesores que impartan docencia de ella, pueden comunicarnoslo en servicio de atención al usuario de los Servicios Informáticos, Apoyo a investigación (91 3 94 47 74).**

Regresión y análisis de la varianza y covarianza.

TTEST Aplicación del test t de Student.
NPAR1WAY Análisis no paramétricos de comparación de grupos.
ANOVA Análisis de la varianza en diseños balanceados. Proporciona también diversos tests de comparaciones múltiples.
GLM Análisis de modelos lineales, usando método de mínimos cuadrados. Proporciona análisis de la varianza univariante y multivariante en diseños balanceados o no, con diversas estructuras de diseño; test de comparaciones múltiples, análisis de la covarianza, regresión simple y múltiple, correlación parcial,...
GLMMOD Front-end para la construcción de modelos a especificar en el procedimiento GLM. Tiene por salida la matriz de diseño.
GENMOD Modelos lineales generalizados, según la definición de Nelder y Wedderburn.
MULTTEST Tests de comparaciones múltiples, incluyendo métodos bootstrap y permutación.
NESTED Análisis de la varianza y covarianza en diseños anidados, especialmente cuando el número de niveles y observaciones es grande.
VARCOMP Cálculo de componentes de la varianza en un modelo lineal general, especialmente con efectos aleatorios.
PLAN Construcción de diseños para experimentos anidados y crossed.
MIXED Modelos lineales mixtos.
REG Análisis de regresión lineal general. Regresión Ridge y análisis de componentes principales incompleto, método por pasos. No posee tantos diagnósticos como el GLM.
ORTHOREG Regresión usando el método Gentelman-Givens.
RSREG Regresión de modelos con una superficie de respuesta cuadrática.
TRANSREG Obtención de transformaciones lineales o no lineales, usando el método de mínimos cuadrados alternado para ajuste a modelos de regresón lineal, correlación canónica y análisis de la varianza. Inclusye transformaciones logarítmicas, rangos, exponenciación y logit.
LATTICE Análisis de la varianza en diseños Lattice.
NLIN Regresión en modelos no lineales.
CALIS Modelos causales. (Path analysis). Análisis de Covarianza de ecuaciones lineales estructurales, en particular, Análisis Factorial Exploratorio y Confirmatorio.
9.1 **ROBUSTREG** Regresiones robustas con presencia de outliers
.

Análisis de datos cualitativos.

FREQ Construcción de tablas de contingencia. Medidas de asociación. Análisis estratificado. Estimación de riesgo relativo.
9.1 **SURVEYFREQ** Tablas de frencuencias n-arias para datos recogidos en encuestas con estimación de porcentajes y sus correspondientes errores estandar.
CORRESP Análisis de correspondencias binarias y múltiples.
PRINQUAL Análisis de componentes principales para datos cualitativos.
CATMOD Ajuste de modelos lineales a funciones de respuesta frecuentistas. Regresión, análisis de la varianza, modelos lineales, log-lineales, regresión logística, medidas repetidas.
LOGISTIC Análisis de regresión logística para respuesta binaria u ordinal mediante máxima verosimilitud. Regresión por pasos. La función de respuesta puede sustituirse por

las funciones 'normit' o log-log. En SAS 9.1, regresión condicional logística (sentencia STRATA) y ordenación de conjuntos de datos (sentencia SCORE)

9.1 **SURVEYLOGISTIC** Análisis de regression para datos de encuestas, incorporando diseños complejos de encuesta: estratificación, conglomerados, ponderaciones, etc… (logit acumulado, logit teneralizado, probit, funciones log-log complementarias,…)

PROBIT Regresión probit, logística y ordinal logística. Variable dependiente dicotómica o policotómica.

Análisis discriminante.

DISCRIM Análisis discriminante, obteniendo funciones discriminantes lineales y cuadráticas. También, métodos no paramétricos.

CANDISC Análisis discriminante canónico. Se asume distribución multivariante normal.

STEPDISC Análisis discriminante por pasos, hacia delante o hacia atrás. Variables continuas.

Análisis cluster.

CLUSTER Análisis cluster jerárquico de observaciones. Distancia euclídea o matrices de disimilaridades (similaridades).

FASTCLUS Análisis cluster para observaciones por el método de las k-medias, usando criterio L_p. Se usa con conjuntos grandes de datos, para la determinación inicial de clusters.

VARCLUS Análisis cluster jerárquico y disjunto de variables, mediante el método de las componentes oblicuas multigrupo.

TREE Muestra la interpretación visual de un análisis cluster jerárquico, mediante dendogramas. Toma como entrada ficheros de salida de CLUSTER o VARCLUS. Genera fichero en el que se indica la pertenencia a cluster en cualquier nivel del árbol del cluster.

MODECLUS Análisis cluster disjunto de observaciones. No paramétrico.

ACECLUS Estimación de matrices de varianzas-covarianzas intra-cluster, sin previo conocimiento de los clusters. Dichas matrices serán la entrada de otros procedimientos de análisis cluster. Util para cluster elípticos alargados.

Análisis de supervivencia.

LIFEREG Análisis de supervivencia. Métodos paramétricos.

LIFETEST Análisis de supervivencia. Métodos no paramétricos.

PHREG Análisis de la regresión de supervivencia, basado en el modelo de azar de Cox.

9.1 **TPHREG** Versión de prueba del procedimiento PHREG que incorpora una variable de clasificación (sentencia CLASS)

Scoring.

STANDARD Standarización de variables.

SCORE Construcción de nuevas variables que son combinación lineal de otras. Se utiliza junto al procedimiento FACTOR u otros.

RANK Asignación de rangos.

Otros procedimientos multivariables.

CANCORR Análisis de correlación canónica.
PRINCOMP Análisis de componentes principales.
FACTOR Análisis factorial, incluyendo rotaciones.
MDS Escalamiento multidimensional.
INBREED Cálculo de la covarianza o coeficientes puros de un pedigree.

9.1 Imputación de datos perdidos

MI y **MIANALYZE** Estrategia de multi-imputación para datos perdidos.

9.1 Análisis de potencia

POWER y **GLMPOWER**. Análisis de potencia y cálculo de tamaños maestrales.

(tb. Power and Sample Size Application surfaces)

Otros.

9.1 **DISTANCE** Cálculo de medidas de distancia, disimilaridad o similiaridad entre observaciones. Métodos paramétricos y no paramétricos de estandarización de variables.

OUTPUT Y TEMPLATE

9.1 **Salida de gráficos con ODS (período de pruebas).** Muchos procedimientos incorporan de forma experimental la salida de gráficos en diversos formatos. Se pueden consultar los ejemplos que aparecen en el manual en los procedimientos estadísticos. En este manual, hay ejemplos en las páginas 69 y 71.

I.3 ESTRUCTURA GENERAL DE UN PROGRAMA SAS®

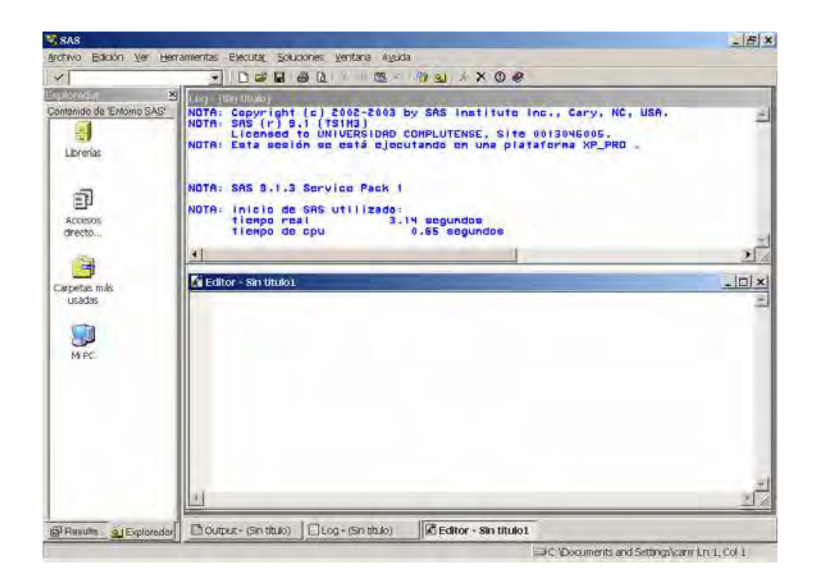
Ejecución de SAS

En WINDOWS

Por lo general realizaremos la ejecución en el entorno de ventanas.

En la ventana *editor avanzado* (o, en su defecto la ventata *program editor*), se incluirá el programa SAS. La salida de resultados, si existe, se incluye en la ventana *output* y la salida de diagnósticos en la ventana *log*. Si se desea almacenar en un fichero el contenido de una ventana, se activa la ventana y se realiza desde *Archivo/Guardar* o *Archivo/Guardar como*.

En general los ficheros de sentencias SAS serán de tipo sas, los de salida de diagnósticos de tipo log y los de salida de resultados de tipo lst.



EN UNIX

La ejecución de un programa SAS se puede hacer de forma interactiva en un entorno de ventanas o bien introduciendo las instrucciones SAS en un fichero y enviando a ejecución ese fichero. Si el segundo nombre es **SAS**, la ejecución se realiza mediante: **SAS** primer-nombre. De dicha ejecución se obtienen los ficheros *primer-nombre.lst* y *primer-nombre.log*, que incluyen respectivamente la salida de resultados si existe y la salida de diagnósticos.

Si el segundo nombre del fichero es distinto de *SAS*, la ejecución se realiza mediante: **SAS** nombre-fichero. De dicha ejecución se obtienen los ficheros *nombre-fichero.lst* y *nombre-fichero.log*, que incluyen respectivamente la salida de resultados si existe y la salida de diagnósticos.

El entorno unix permite también la ejecución en BATCH, enviando la ejecución a una de las colas de la máquina.

Estructura de un programa SAS

Conjunto de datos.

Es el conjunto de datos que se desea analizar. Puede incluirse:

- ♣[2]En el fichero de sentencias SAS,
- ♣Permanecer en otro fichero externo. El fichero externo puede contener:
  - ♣Los datos en ASCII
  - ♣Los datos en una base de datos, por ejemplo EXCEL
  - Los datos en un conjunto de datos SAS o SAS-FILE permenente. Los conjuntos de datos SAS son ficheros internos del entorno SAS. Son permanentes si existen en el disco de una sesión a otra de SAS.
- Ser introducidos directamente en SAS .

Sentencias de control o programa SAS®.

[2] **El símbolo ♣ indica las opciones más usuales o las que corresponden a este manual de usuario.**

Son las sentencias que indican al SAS la procedencia de los datos y sus características, la necesidad de realizar transformaciones a los mismos, el tipo de análisis que se desea realizar y qué resultados obtener. Un programa SAS está formado por sentencias SAS que se encuentran agrupadas en pasos DATA o pasos PROC, quedando determinada la terminación de un paso por el comienzo de otro.

En los pasos DATA se crean o modifican conjuntos de datos SAS que serán procesados por los pasos PROC. Asimismo, pueden utilizarse para crear ficheros externos de datos o generar informes mediante programación.

Los pasos PROC teniendo como entrada un conjunto de datos SAS, son los que realizan los análisis relativos a los mismos. Comienzan por la palabra PROC seguido del nombre del procedimiento.

Ejemplo de programa SAS®:

```
/*              Título en la salida de resultados         */
      TITLE  'Este es un programa que sirve de ejemplo';

/*              Creación de formatos de escritura         */
      PROC  FORMAT;
            VALUE  sex  1='varon'  2 = 'mujer';

/*          Creación del SAS-FILE a analizar y entrada de datos  */
      DATA  analizo;
      INFILE  'morfo.datos';
      INPUT  id  $4.  (tamaño  altura  peso)  (5.2)  sexo  2.;
      ARRAY  algun{2}  tamaño  altura;

/*          Identificación de valores perdidos o no observados  */
      DO  I = 1  TO  2;
            IF  (algun{I} = 99.99)  THEN  algun{I} = . ;
      END;
      IF  (sexo = 9)  THEN  sexo = . ;

/*              Transformación de variables               */
      bmi = (peso**2) / (altura*100);

/*              Asignación de formatos de escritura       */
      FORMAT  sexo  sex. ;

/*              Eliminar la variable I del SAS-FILE       */
      DROP I ;

/*              Análisis realizados sobre el SAS-FILE creado  */
      PROC  SORT  DATA = analizo ;
      BY  sexo;

      PROC  PRINT;

      PROC  MEANS;
      BY  sexo;

      PROC  TTEST  COCHRAN ;
      CLASS  sexo;
      VAR  tamaño--bmi ;
```

En este ejemplo, se especifica un **título** que aparecerá en la salida de resultados. Posteriormente, se crea un **formato** de escritura que podrá asignarse a una variable. En el paso **DATA**, se crea un **conjunto de datos SAS o SAS-FILE**, se especifica el **fichero externo** donde se encuentran los datos, cuáles son las **variables** de las que consta, se especifican **valores de no observación**, se **crea una variable** transformación de otras en entrada y finalmente se asigna el **formato** previamente creado, a la variable sexo para que en la salida de resultados aparezcan los descodificados de la misma. Una vez creado el **SAS-FILE**, se analizará en los **procedimientos o PROC** sucesivos. **PROC SORT** ordena el SAS-FILE por los valores de la variable sexo, **PROC PRINT** produce un listado de los datos, **PROC MEANS** realiza una estadística básica de los mismos categorizada por los valores de sexo, **PROC TTEST** calcula el estadístico t de Student para la comparación de sexos. Se requieren también las aproximaciones de Cochram y Cox para el caso de heterocedasticidad.

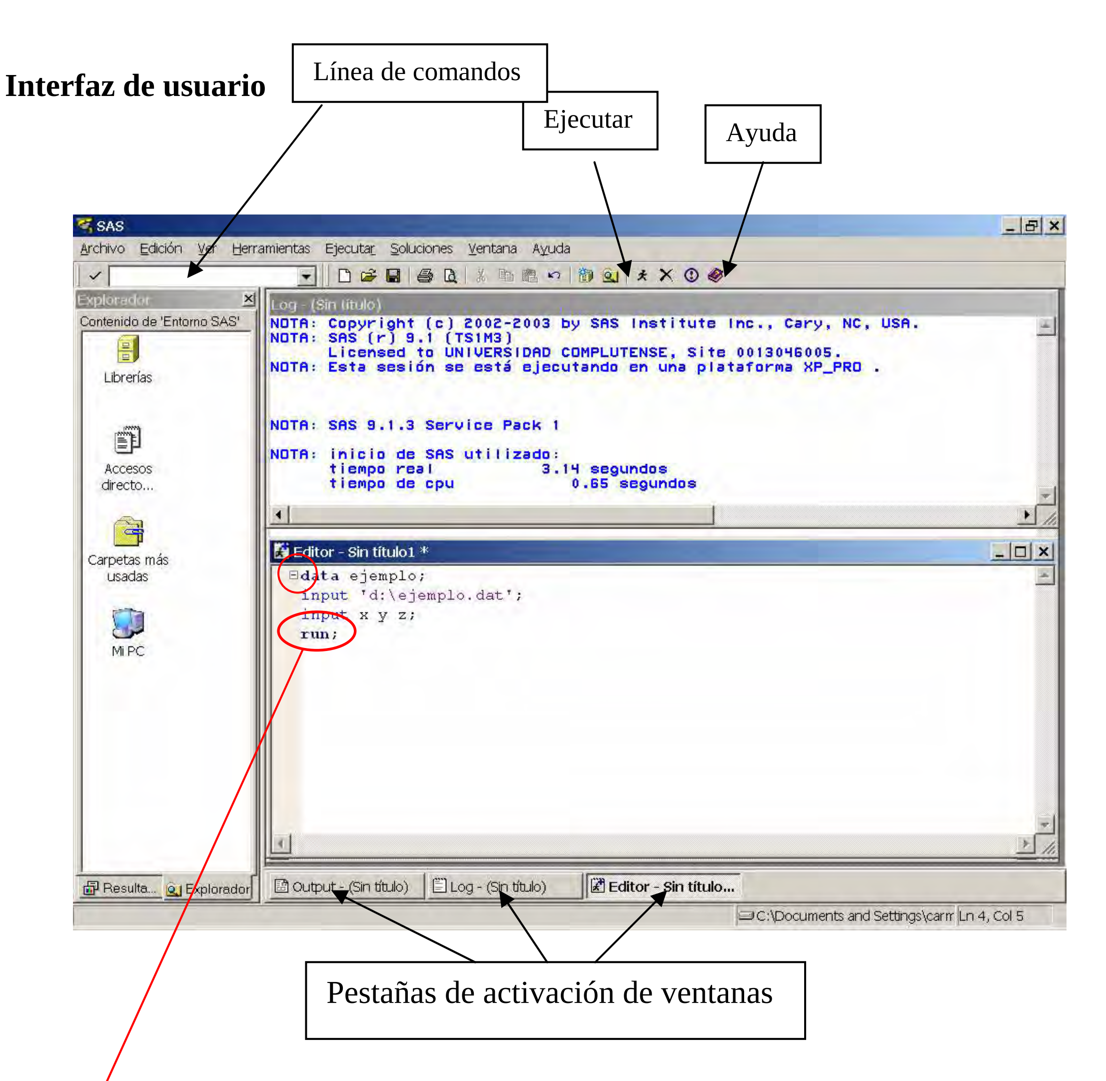
Salida de resultados.

Es el resultado del análisis sobre el conjunto de datos según las pautas reseñadas por las sentencias de control. Si la ejecución se realiza de forma no interactiva aparecerá en un fichero con último nombre **lst** y si es de forma interactiva en la ventana *OUTPUT*.

Salida de diagnósticos.

Durante la ejecución, SAS va realizando diagnósticos sobre los pasos que procesa. En ellos se incluyen los errores de sintaxis, el tiempo de ejecución y memoria utilizadas, las páginas de la salida de resultados en que aparecen los resultados correspondientes a cada paso, etc... Si la ejecución se realiza de forma no interactiva aparecerá en un fichero con último nombre **log** y si es de forma interactiva en la ventana LOG. Si se desea borrar el contienido de la ventana Log, se selecciona la misma con el botón izquierdo del ratón y se pulsa la tecla que aparece en la barra del menú.

Interfaz de usuario



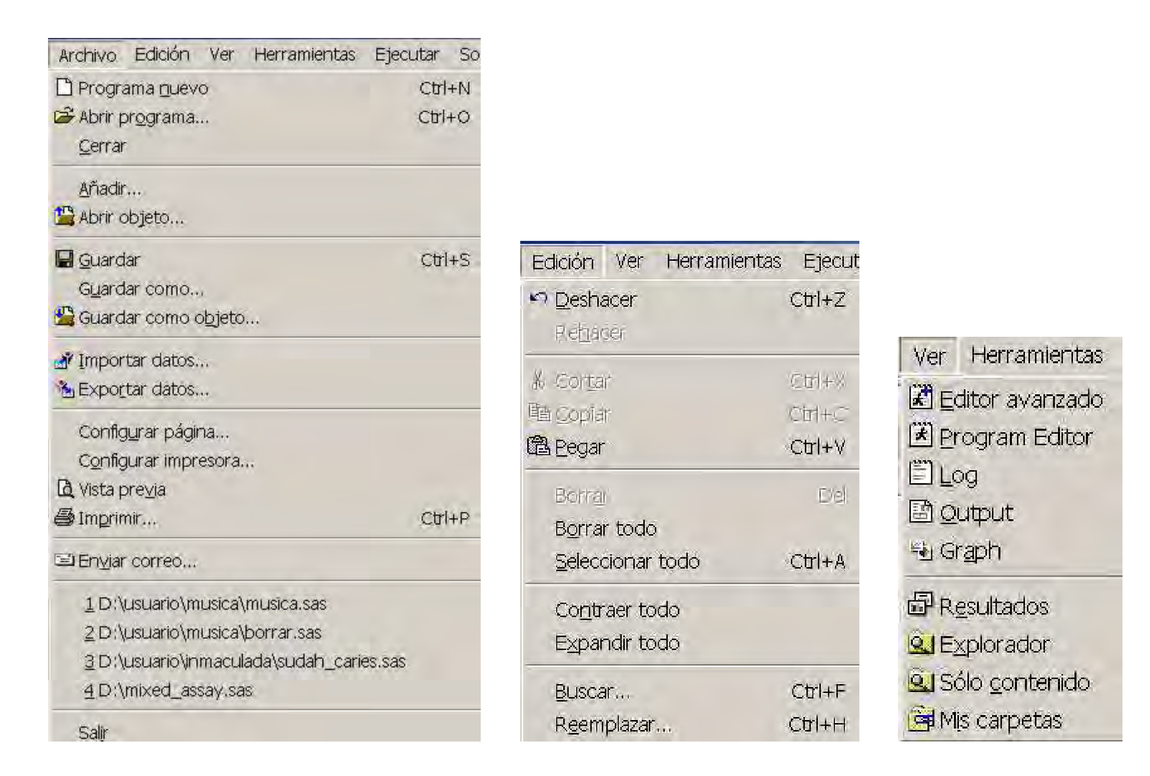
Cuando la ventana ***editor avanzado*** (o *program editor*) está activa, el icono de ejecución permite la ejecución de partes del contenido de la ventana correspondiente sin más que seleccionar esa parte con el ratón. Si no se ha seleccionado un texto, se ejecuta todo el contenido de la ventana.

La sentencia run; marca el final de un paso data o paso proc. El final de los pasos data o proc queda marcado igualmente por el inicio de otro paso data o proc. Es conveniente incluir la sentencia run; al final de un bloque que se envía a ejecución, aunque es innecesaria incluirla entre pasos data o proc.

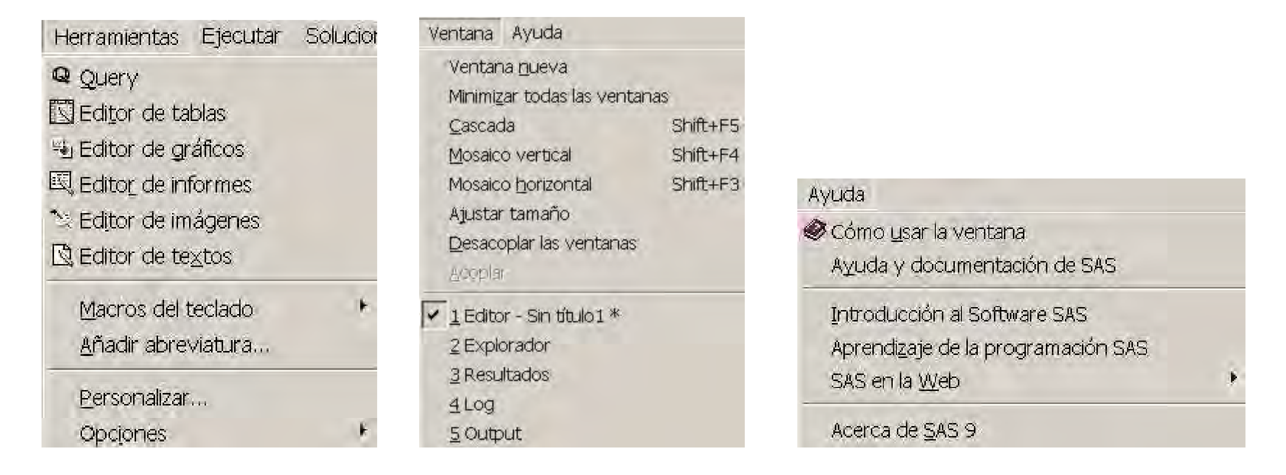
Se recomienda el uso de la ventana *editor avanzado* en lugar de la ventana *program editor* (presente en versiones anteriores de SAS como ventana PROGRAM) ya que:

- El contenido de la ventana *editor avanzado* permanece en la ventana despues del envío de sentencias a ejecución.
- Es sensible al lenguaje mostrando diversos colores en las palabras.
- Tiene los símbolos – y + que permiten colapsar o delplegar conjuntos de sentencias.

Menues:

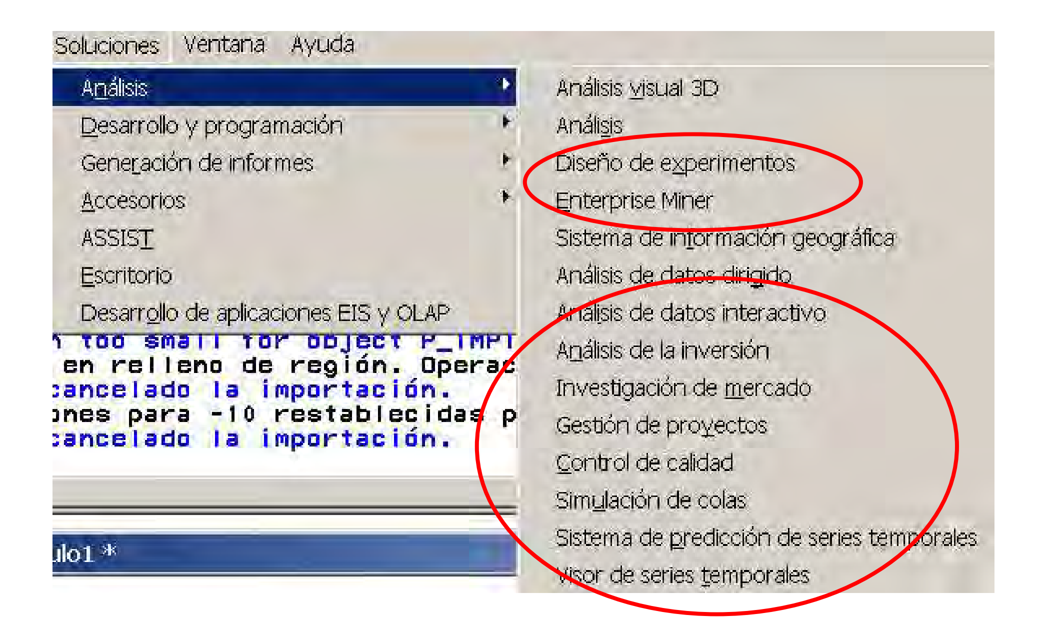


El menu **Ver** permite abrir algunas ventanas si estas se han cerrado: *editor avanzado*, *program editor, log, output, graph, resultados*. La opción ***solo contenido*** activa la ventana ***explorador***. La opción ***explorador*** activa otra ventana de exploración, que no presentaremos en este manual.



La pestaña ***explorador*** (véase algo más adelante) permite ver y gestionar las librerías y ficheros SAS y carpetas (shortcut) de ficheros no SAS. Permite crear librerías y ficheros SAS, abrir, copiar, mover, borrar ficheros, etc… Una librería de SAS se asocia a un directorio y contiene tablas o SAS-files (ademas de Queries y Catálogos (de formatos, gráficos, ...)).

La pestaña ***resultados*** permite navegar entre las distintas salidas de los procedimientos que se encuentran en la ventana *output*.



La ejecución de SAS en el entorno de ventanas

La ejecución se ordena activada la ventana ***editor avanzado*** (o *program editor*) seleccionando con el ratón el conjunto deseado de sentencias. Si se desea ejecutar todas las sentencias de la ventana no es necesario seleccionarlas.

El final de un paso data o un procedimiento se indica por el comienzo de otro. Para indicar que el último paso data o proc (procedimiento) enviado a ejecución ha finalizado, debe indicarse el comando o sentencia ***run;*** .

Al enviar a ejecución código SAS, se muestran en la ventana ***log***, el diagnóstico de la ejecución. Si se producen resultados, éstos aparecen en una ventana ***output*** que se muestra automáticamente.

La ventana *Explorador*

La ventana ***explorador*** permite ver y gestionar las librerías y ficheros SAS (SAS-FILES) y carpetas (shortcut) de ficheros no SAS. Permite crear librerías de datos y ficheros SAS, abrir, copiar, mover, borrar ficheros, etc… Para visualizar la pantalla *explorador* si se ha cerrado previamente: Menu Ver/ Opción solo contenido.

Una librería de SAS se asocia a un directorio y contiene tablas o ficheros SAS (ademas de Queries y Catálogos que exceden el contenido de esta guía).

Las librerías inicialmente asociadas en SAS son:

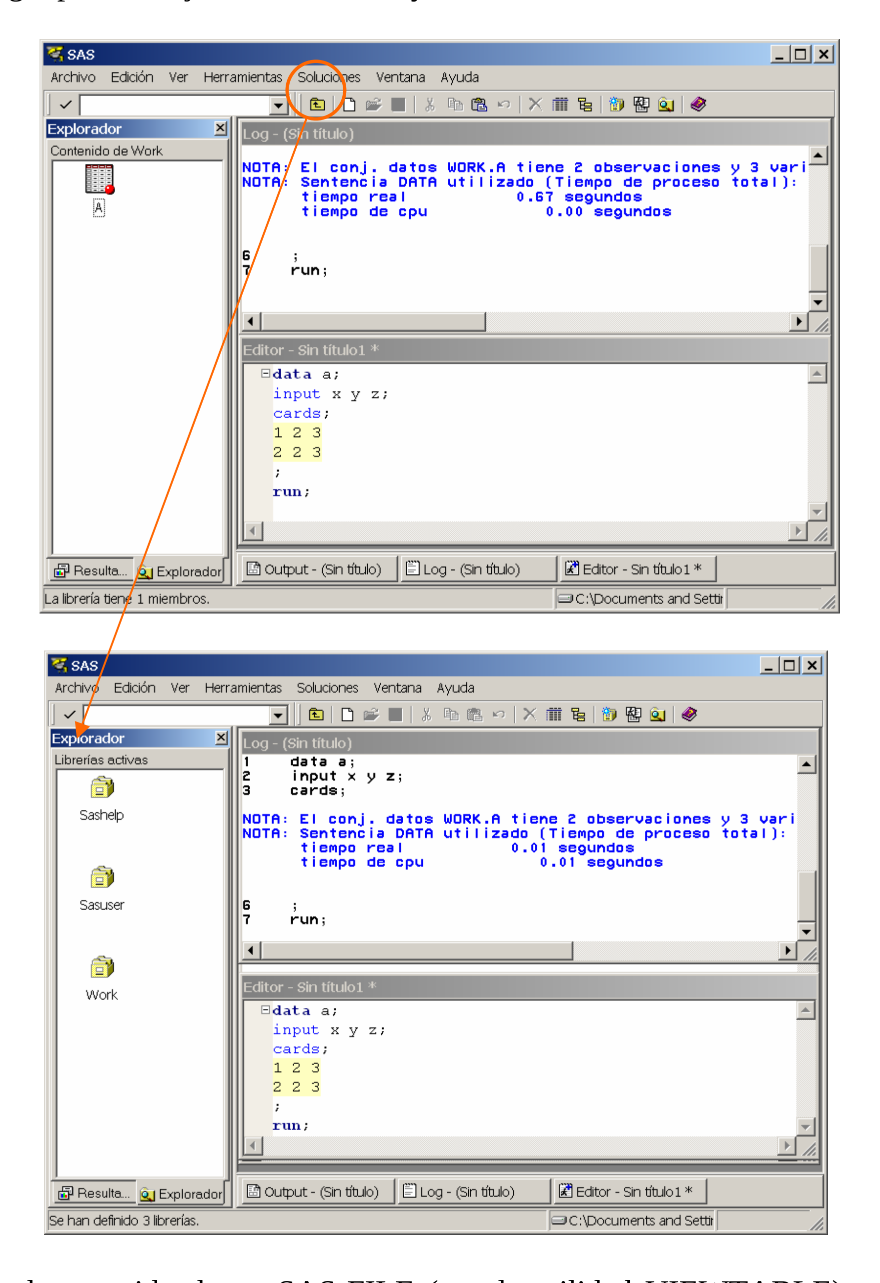
SASHELP   Contiene SAS-FILES de ejemplo que son utilizados en los ejemplos de ayuda del programa SAS.
MAPS   Contiene SAS-FILES de ejemplo acerca de países
SASUSER
WORK   Se incluyen los SAS-FILES temporales, es decir, los que se crean en una sesión SAS y se eliminan al terminar la sesión.

Las operaciones realizadas desde la ventana *explorador* no generan código SAS.

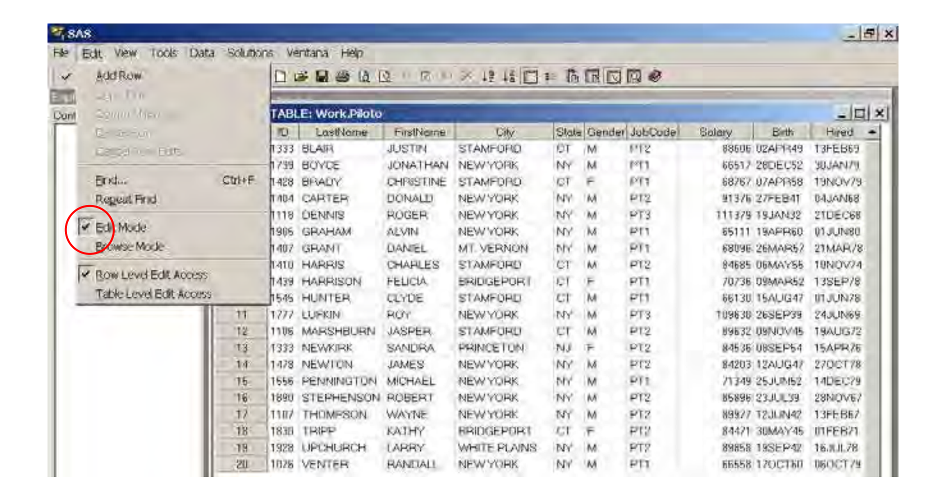
Algunas utilidades de la ventana *explorador*:

Navegar por las distintas librerías y los distintos SAS-FILES
Ver el contenido de los SAS-FILES y, en modo edición, cambiar su contenido.
Crear nuevas librerías y nuevos SAS-FILES y catálogos de estas librerías.
Copiar, mover, borrar librerías, SAS-FILES o ficheros, etc…

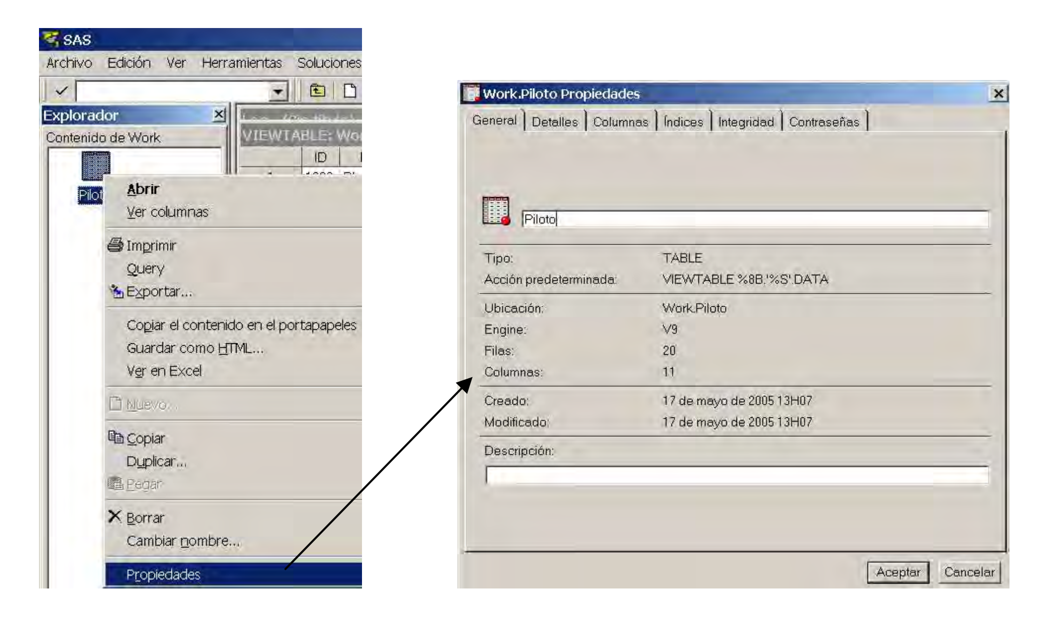
Se puede navegar por el conjunto de librerías y SAS-FILES:



Se puede ver el contenido de un SAS-FILE (con la utilidad VIEWTABLE) realizando una doble pulsación del botón izquierdo sobre el nombre del SAS-FILE. Para modificar datos, una vez visualizado el SAS-FILE, se debe seleccionar en el menu Edit, la opcion Edit Mode:

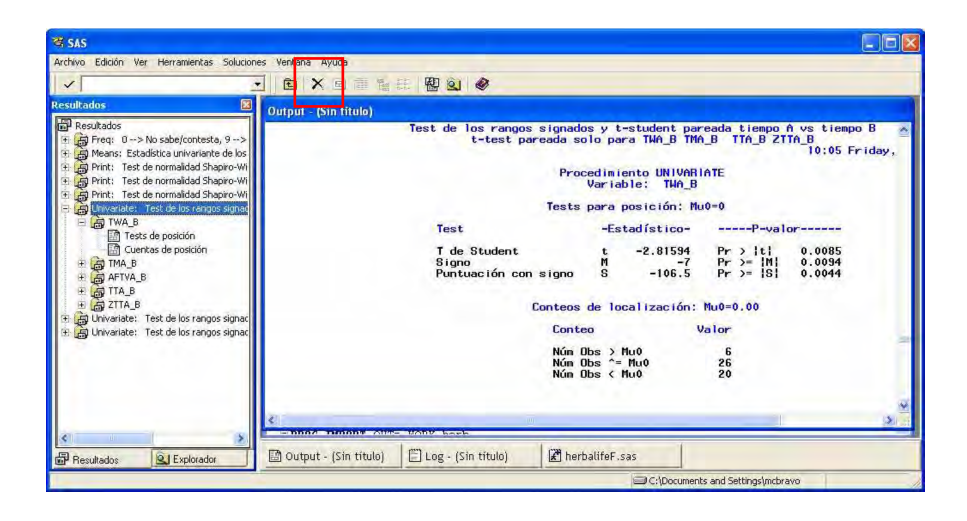


Se pueden ver las propiedades de los SAS-FILES:



La ventana *Resultados*

La ventana resultados permite navegar por los resultados de los procedimientos, así cómo la eliminación parcial de estos resultados.



Si se desea borrar el resultado de algún procedimiento de la ventana Output, se debe seleccionar en la ventana Resultados el procedimiento que se desea eliminar y pulsar el icono que aparece en la barra del menú. Si se desea eliminar todo el contenido de la ventana Output basta con seleccionar la ventana Output pulsando el botón izquierdo del ratón sobre ella y seguidamente pulsar el icono que se encuentra en la barra del menú.

I.4 NORMAS GENERALES DE SINTAXIS DE UN PROGRAMA SAS

-No se tiene en cuenta la posición de las palabras por columnas fijas, pudiéndose dejar tantos espacios o líneas en blanco como se quiera.

-Cada sentencia SAS comienza por una palabra clave y debe terminarse por un punto y coma (;). No existe restricción en el número de líneas a ocupar por una sentencia ni en el número de sentencias en una misma línea.

-La mayoría de las sentencias SAS tienen la posibilidad de especificar opciones, bien mediante una palabra clave, bien mediante una palabra clave seguida del signo = y de un valor dado por el usuario.

-La separación de elementos en una sentencia se realiza mediante uno o más espacios en blanco. Asimismo, pueden existir líneas en blanco entre elementos de una misma sentencia.

-La única restricción de posición es la de la sentencia nula (;) al finalizar las líneas de datos si éstos se encuentran en el programa SAS. Dicha sentencia nula debe encontrarse en una nueva línea y en la primera posición.

Elementos propios de la sintaxis del SAS®.

= se utiliza en:
- -sentencias de asignación.
- -opciones de sentencias SAS, separando una palabra clave SAS del valor que el usuario le asigna.

' (apóstrofe) encierra:
- -un título en las sentencias TITLEn
- -un nombre de fichero externo en las sentencias INFILE, FILE, FILENAME, %INCLUDE.
- -una descodificación en el PROC FORMAT, en la sentencia VALUE.
- -un nombre de directorio en la sentencia LIBNAME.
- -valores de variables de tipo carácter.

/ (slash) en ocasiones, separa en una sentencia SAS, la lista de opciones.

( ) (paréntesis) agrupa términos o expresiones arimetico-lógicas o carácter.

{ } (llaves) encierra el índice de los elementos de un array.

alt1-alt3 (por ejemplo) indica las variables de nombre alt1, alt2 y alt3.

vari--varf indica todas las variables comprendidas entre vari y varf.

/* comentario */
ó *** comentario ;** incluye comentarios en el programa SAS.

$ usado en la designación de *macro-variables*.
Indicación de variables carácter.

% usado en la designación de *macros* y en la sentencia %INCLUDE.

Características de los nombres aportados por los usuarios en SAS®.

Estos pueden ser nombres de variables, SAS-FILEs, formatos, *macros*, etc...

Los nombres aportados deben constar de 1 a 32 caracteres alfanuméricos o _, no pudiendo comenzar por un número.

Los nombres de formatos no pueden terminar en número y si se atribuyen a variables carácter deben comenzar por $. Asímismo, SAS se reserva el uso de determinados nombres, como algunos que empiezan y terminan por _, así: _N_, _ERROR_, _I_,...

Convenciones de notación en esta guía.

[ ] (corchetes)	encierra algo que es opcional.
| (barra)	indica una disyunción excluyente.
... (puntos suspensivos)	indican la posible repetición de lo inmediatamente anterior.
MAYUSCULA	indica palabras clave de SAS.
minúscula	indica palabras que debe proporcionar el usuario.

Observación.

En las sentencias que se muestren en esta guía, se especificarán algunas opciones posibles. Para todas ellas existen muchas otras posibilidades.

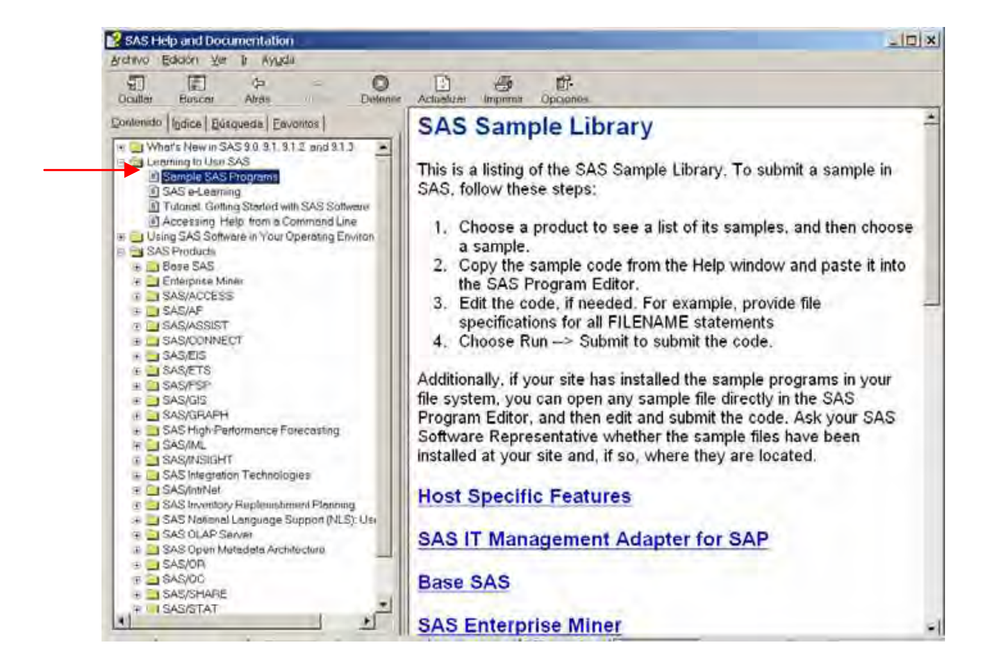
Documentación de SAS

La documentación, en forma de manual, de SAS 9.1.3 se puede consultar dentro del producto a traves de la opción de menú ayuda → Ayuda y documentación SAS, en la pestaña contenido
.

Además, en la url de SAS:
http://support.sas.com/documentation/onlinedoc/sas9doc.html (y más concretamente en http://support.sas.com/documentation/onlinedoc/91pdf/index_913.html)
se encuentra la misma documentación acompañada de ficheros en pdf.

Ayuda de SAS®

En el programa SAS existe una ayuda bastante extensa. Para acceder a la ayuda hay que teclear el icono de ayuda que está simbolizado por un libro o bien acceder desde la opción de menú *Ayuda*. La ayuda incluye información acerca de la sintaxis de SAS e información acerca de los procedimientos y su sintaxis. La ayuda incluye además ejemplos de programas en SAS. A partir de la opción de menú *Ayuda/Ayuda y documentación*, veamos una de las ventanas, la de acceso a los programas de ejemplo de SAS.



Ayudas acerca de los procedimientos y su sintaxis:

SAS Help and Documentation

The CANDISC Procedure

Syntax

The following statements are available in PROC CANDISC.

PROC CANDISC < options > ;
  CLASS variable ;
  BY variables ;
  FREQ variable ;
  VAR variables ;
  WEIGHT variable ;

The BY, CLASS, FREQ, VAR, and WEIGHT statements are described after the PROC CANDISC statement.

PROC CANDISC Statement

BY Statement

CLASS Statement

FREQ Statement

VAR Statement

WEIGHT Statement

Previous | Next | Top of Page

Copyright © 2003 by SAS Institute Inc., Cary, NC, USA. All rights reserved.

SAS Help and Documentation

The CANDISC Procedure

Example 21.1: Analysis of Iris Data Using PROC CANDISC

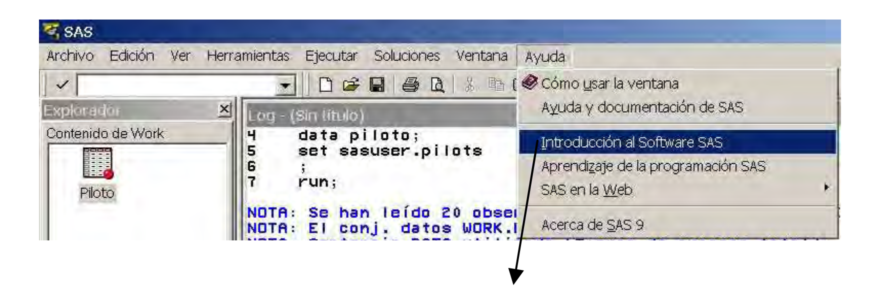
The iris data published by Fisher (1936) have been widely used for examples in discriminant analysis and cluster analysis. The sepal length, sepal width, petal length, and petal width are measured in millimeters on fifty iris specimens from each of three species: *Iris setosa*, *I. versicolor*, and *I. virginica*.

This example is a canonical discriminant analysis that creates an output data set containing scores on the canonical variables and plots the canonical variables. The following statements produce Output 21.1.1 through Output 21.1.7:

```
proc format;
   value specname
      1='Setosa    '
      2='Versicolor'
      3='Virginica ';
run;

data iris;
   title 'Fisher (1936) Iris Data';
   input SepalLength SepalWidth PetalLength PetalWidth
         Species @@;
```

Si se desea una ayuda acerca de la programación SAS, debe activarse (desde fuera de la ayuda), la opción de menú *Ayuda/Introducción al Software SAS*





Los tutoriales a los que hace referencia la primera ventana de SAS para nuevas instalaciones, son los que se presentan en la ventana que mostramos.

II. ENTRADA DE DATOS EN SAS[3]

II.1 CREACION DE CONJUNTOS DE DATOS SAS® (SAS-FILE).

Como ya se ha dicho anteriormente, es necesaria la creación de un SAS-FILE para el procesamiento de cualquier conjunto de datos por los procedimientos de SAS.

Un SAS-FILE es un fichero con una estructura especial creada por SAS y almacenado en binario. Este fichero almacena, además de los datos, toda la información relativa a las variables, sus nombres, etiquetas, valores desconocidos, tipo de los datos, etc...

Los SAS-FILEs pueden ser temporales a la ejecución del programa SAS o permanentes. Los SAS-FILEs temporales se almacenan automáticamente en una librería de datos WORK. Los SAS-FILEs permanentes se almacenan en una librería de datos que el usuario debe especificar con el comando LIBNAME o con la interfaz de usuario de SAS . El nombre del SAS-FILE se compone, en este caso, del nombre de la librería, seguido de un punto y del nombre del fichero donde se almacena el SAS-FILE.

A) Creación de un SAS-FILE, por entrada de datos externos a SAS

Éste puede obtenerse:

a) En un paso DATA, por entrada de datos del usuario contenidos en un fichero externo en ASCII (Véase II.A (página 25) y II.B (página 33)).

b) Mediante el entorno gráfico de SAS, por entrada de datos ficheros externos en ASCII, (Véase II.B (página 33)) o por importación[4] de datos de otros formatos (Ej: Excel). (Véase II.C (página 41)).

c) En un paso DATA, por inserción de datos en el código SAS (Véase II.A (página 25)).

d) Mediante introducción de datos directamente en SAS (véase II.D (página 44)).

---

[3] **En II.B (página 33) también se incluye la exportación de datos**

[4] Procedimiento IMPORT

B) Creación de un SAS-FILE, por otros medios

Éste puede obtenerse:

a) En un paso DATA, por entrada de datos de otros SAS-FILEs creados con anterioridad. Véase capítulo IV (página 57)).

b) En un paso DATA, mediante programación, haciendo uso de las herramientas de SAS: asignación, operaciones, funciones... (Véase capítulo III (página 45)).

c) Como salida de un procedimiento

C) Creación de un SAS-FILE en un paso DATA (incluye opciones de A) y B))

Como veremos más adelante y a modo de resumen, en el caso de crearse el SAS-FILE en el paso DATA, las sentencias necesarias en un paso DATA son:

-En todos los casos, la sentencia DATA para indicar el comienzo del paso.

-Si se leen datos ASCII: las sentencia INFILE (si los datos están en un fichero externo) ó la sentencia CARDS (si los datos están en el código SAS) y la sentencia INPUT en ambos casos.

-Si se leen datos de un SAS-FILE preexistente: la sentencia SET o la sentencia MERGE o la sentencia UPDATE, sin mezclar sentencias de distinto nombre en el mismo paso DATA.

Características tabulares de los SAS-FILE.

Los SAS-FILEs son conjuntos de datos específicos de SAS que contienen, en formato binario, matrices de casos u observaciones (filas) por variables (columnas). Cada caso está compuesto por los valores de las variables que toma dicha observación.

La tabularización, sin embargo, no requiere que los ficheros de datos aportados por el usuario en entrada, sean rectangulares, ya que SAS tiene posibilidad de entrada de datos en muchas estructuras posibles.

II.2 LIBRERÍAS DE DATOS SAS®

Una librería de datos SAS es un conjunto de SAS-FILEs en un directorio. Un mismo directorio puede incluir diferentes librerías SAS. Cada librería SAS tiene asociados el directorio en el que se encuentra y una referencia que coincide con el nombre de la librería.

Por defecto, los SAS-FILES temporales que es crean en un sesión se almacenan en una librería de datos temporal (la librería WORK) que al terminar la sesión desaparece.

En general, y en relación con SAS-FILEs, sólo se hace referencia a una librería SAS en el caso de que se desee crear un SAS-FILE permanente o leer de un SAS-FILE permanente.

SAS crea dos librerías de datos: SASHELP y SASUSER, ésta última con SAS-FILES de ejemplo.

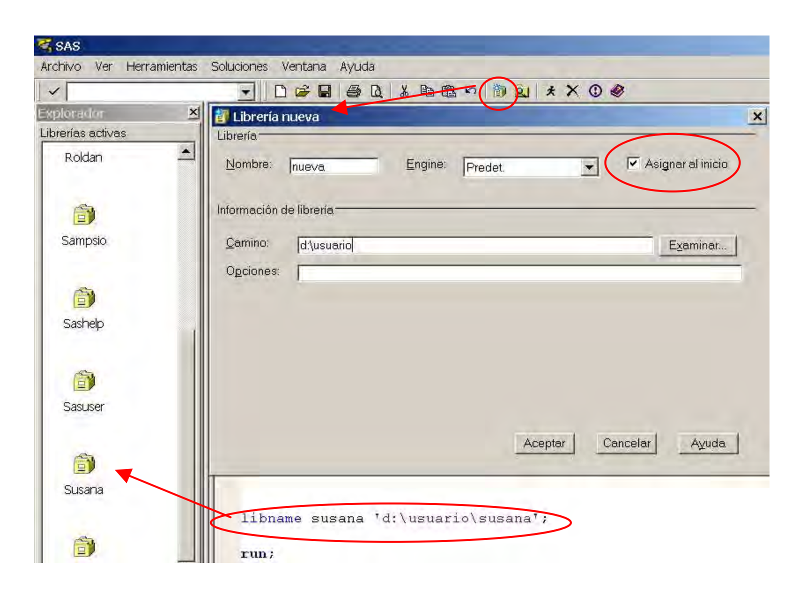
Se pueden crear librerías de datos SAS de dos formas alternativas:

- El entorno gráfico de SAS
- La sentencia LIBNAME

Creación de librerías de datos SAS con el entorno gráfico de SAS

Se pueden crear librerías de datos SAS con el entorno gráfico de SAS de la siguiente forma:

- Activada la ventana *explorador*[5] y sin seleccionar una librería ni estar en el interior de ninguna de las librerías: opción de menu *Archivo/Nuevo/Librería*
- O bien con el icono que aparece en la imagen siguiente.



En esta imagen se crean dos librerías:

- Y la librería nueva se crea con el icono . La opcion ‘Asignar al inicio' hace posible que la librería creada sea visible en otra sesión de SAS.
- La librería usuario se ha creado con la sentencia LIBNAME (véase en el siguiente epígrafe);

Las librerías creadas con el entorno gráfico no generan el código de la sentencia LIBNAME.

Sentencia LIBNAME

La sentencia LIBNAME es externa a un paso DATA o PROC y debe especificarse en el programa con anterioridad al uso de la librería que crea.

Sintaxis:

```
LIBNAME  ref-librería  'directorio';
```

ref-librería: nombre que se da a la librería
directorio: directorio de la librería. Dicho directorio debe existir previamente.

Ejemplo:

```
LIBNAME susana ‘d:\usuario\susana\datos_sas’;
```

[5] **La ventana *explorador* se presenta en la página 15**

```
DATA susana.colesterol;
INFILE ‘d:\usuario\susana\datos_ascii\colesterol.txt’;
INPUT id genero edad col hdl ldl ;

PROC PRINT DATA=susana.colesterol;
RUN;
```

En este ejemplo se lee de un fichero externo de datos llamado `colesterol.txt` **que se encuentra en la carpeta** `d:\usuario\susana\datos_ascii` **y se crea un SAS-FILE llamado** `colesterol` **de la librería** `susana` **(que se encuentra en la carpeta** `d:\usuario\susana\datos_sas`**). La forma de referenciar el SAS-FILE, al ser permamente es** `susana.colesterol`.

Nota Importante

Las librerías creadas:

- Con la sentencia LIBNAME
- O, Activada la ventana *explorador*, opción de menu *Archivo/Nuevo/Librería*
- O, con el icono de creación de librerías sin activar la opción ‘Asignar al inicio’…

… en una sesión de SAS, no son visibles en la ventana *explorador* de otra sesión SAS, aunque existan en los directorios correspondientes los SAS-FILES permanentes. Para hacer visibles las librerías de datos SAS (y así que sean visibles los SAS-FILES correspondientes) basta con crear (asignar) las librerías en la nueva sesión[6].

II.A. ENTRADA DE DATOS Y DECLARACION DE VARIABLES en el PASO DATA

En esta sección, se especifica la entrada de datos del usuario contenidos en un fichero externo en ASCII o contenidos en el programa SAS. En la sección siguiente se muestra la entrada de datos de ficheros externos en **ASCII** con el entorno gráfico de SAS .

Al crear un SAS-FILE, los datos de entrada pueden encontrarse en un fichero externo o en el fichero del programa SAS. En ambos casos son necesarias las sentencias DATA e INPUT. En el primer caso se utilizará la sentencia INFILE para indicar el fichero externo y en el segundo la sentencia CARDS, para indicar el comienzo de los datos; que una vez finalizados irán seguidos por la sentencia nula (;). Las secuencias de sentencias obligatorias, son respectivamente:

(si datos en fichero externo)   (si datos en programa SAS)

```
DATA                          DATA
INFILE                        [ INFILE  CARDS ; ]   (optativa)
INPUT                         INPUT
(otras sentencias SAS)(otras sentencias SAS)
                              CARDS;
                              (datos)
```

[6] Si se desea que SAS acceda automáticamente a las librerías al iniciar la sesión, basta con incluir las correspondientes sentencias LIBNAME en el fichero autoexec.sas que se encuentra en el directorio donde se encuentra el ejecutable SAS.exe. El fichero autoexec.exe se ejecuta al iniciar la primera sesión SAS. Para otras sesiones simultáneas en el mismo ordenador, no se ejecuta.

;

Ejemplos básicos de paso DATA.

A) Si los datos se encuentran en un fichero externo:

```
DATA ejemplo;
INFILE 'ejemplo.dat';
INPUT pac sexo altura hdl ldl htamin htamax;
```

B) Si los datos se encuentran en el programa SAS

```
DATA ejemplo;
INPUT pac sexo altura hdl ldl htamin htamax;
CARDS;
1 1 150 121 80 14.0 8.0
2 1 164 100 70 17.0 9.5
3 2 175 98 64 16.5 8.5
4 2 180 110 70 14.0 6.5
…………..
;
```

Modo de ejecución de un paso DATA.

En general y en el caso que se lean observaciones, un paso DATA en el que no hay sentencias de variación de flujo, se ejecuta tantas veces como observaciones se leen. Leída una observación, se procesan sobre ella todas las sentencias ejecutables del paso DATA de forma secuencial. Terminada la ejecución del paso DATA al llegar a la última sentencia del mismo, y una vez incluida la observación en el SAS-FILE, SAS comienza una nueva ejecución del paso DATA para la siguiente observación y así sucesivamente.

Variaciones a esto:

-No hay sentencia INPUT, es decir, los datos son creados en el paso DATA.
-Existen sentencias ejecutables anteriores a la sentencia INPUT. Dichas sentencias se ejecutarán antes de cada lectura de datos.

SENTENCIA DATA.

La sentencia DATA:

-Indica el inicio del paso DATA y da nombres a los SAS-FILEs que se crean.

-Los SAS-FILEs pueden ser temporales a la ejecución del programa SAS o almacenarse en un fichero de forma permanente para su uso en otros programas SAS.

-Un SAS-FILE temporal puede identificarse por un nombre aportado por el usuario o bien por un nombre dado por defecto por SAS: _DATA1_ para el primero del programa, _DATA2_ para el segundo y así sucesivamente.

-La creación de un SAS-FILE permanente requiere la creación precedente de una librería de datos SAS (almacén de SAS_FILEs) mediante la sentencia LIBNAME o el entorno gráfico (véase página 23). El nombre de los SAS-FILEs de una libería de datos, se identifica por el nombre de la librería, seguido de un punto (.) y seguido de otro nombre.

-Puede no crearse un SAS-FILE en la ejecución de un paso DATA si se especifica, en lugar de un nombre, la palabra _NULL_. Esto puede tener utilidad, por ejemplo, si el paso DATA es utilizado para salida de ficheros externos. (Véase capítulo III).

Sintaxis:

DATA _NULL_ ;

ó DATA nombre | nombre-lib.nombre [(opción-SAS-file ...)] ... ;

En el caso de creación de un SAS-FILE permanente, debe especificarse *nombre-lib.nombre*. Además, previa a la sentencia DATA, debe crarse la librería *nombre-lib*, bien mediante la sentencia:

LIBNAME nombre-lib 'directorio-lib';

Bien mediante el entorno gráfico.

El SAS-FILE creado se almacena en disco en el directorio *directorio-lib*, en un fichero llamado *nombre.sas7bdat.*

Ejemplo:

```
LIBNAME censo ‘c:\datoscensales\sasfiles’;

DATA censo.año2000;
INFILE ‘c:\datoscensales\censob.dat’;
INPUT id edad sexo edad profesión estudios;
```

En este ejemplo estamos leyendo datos del fichero externo censob.dat en ASCII que se encuentra en el directorio c:\datoscensales. Creamos un SAS-file permanente, **censo.año2000**, que se encuentra almacenado en el directorio c:\datoscensales\sasfiles con el nombre de fichero año2000.sas7bdat.

Algunas opciones de SAS-FILE, a las que retornaremos una vez que el capítulo IV presente el significado de las sentencias correspondientes en el paso DATA, son:

FIRSTOBS = n	Indica comienzo de proceso a partir de la observación n-ésima.
OBS = n	Indica el procesamiento de n observaciones.
WHERE = (expresion)	Indica selección de observaciones que cumplen la expresión.
TYPE = tipo	Indica tipo de datos: DATA, CORR, COVA...
DROP = var ...	Exclusión de variables en el SAS-FILE.
KEEP = var ...	Mantenimiento de variables en el SAS-FILE.
RENAME = (nombreviejo = nombrenuevo ...)	Renombra variables.

SENTENCIA INFILE.

La sentencia INFILE:

-Identifica como entrada, un fichero externo de datos en ASCII. (salvo INFILE CARDS; , que, además de ser optativa sólo se especifica si los datos se encuentran en el programa SAS). Se debe especificar el nombre del fichero[7] en disco, encerrado por comillas ('). Es conveniente especificar la ruta completa del fichero.

Ejemplo:

```
      DATA salud;
INFILE ‘d:\ejemplos\salud.dat’;
```

-Describe las características del fichero externo donde se encuentran los datos.

-Especifica la forma de lectura de la sentencia INPUT.

-Indica modo de actuación cuando se pretende leer más allá de la longitud del fichero.

Sintaxis:

INFILE   CARDS | 'nombre-fichero' [opcion ...] ;

Posición:

-Debe ir precediendo la sentencia INPUT correspondiente.

Observaciones:

-Puede haber más de una sentencia INFILE si se leen datos de varios ficheros externos. Una sentencia INPUT, en este caso, leerá del fichero reseñado en el último INFILE.

-Una observación puede leerse en una o más líneas. Asimismo, en una línea puede haber una o más observaciones.

---

[7] Se puede identificar el fichero mediante una referencia o nombre lógico que identifica el fichero. Cuando se especifica el nombre de un fichero es conveniente especificar la ruta completa del fichero. En este caso de especificación de una referencia, debe existir antes del paso DATA una sentencia FILENAME. De esta forma queda permitida la lectura de ficheros almacenados en otros dispositivos (cinta, ...)

FILENAME   referencia   'nombre-fichero' | especif-dispositivo ;

Ejemplo:

Se muestra la equivalencia con el ejemplo presentado, mediante referencia:

```
FILENAME datos ‘d:\ejemlos\salud.dat’;
DATA salud;
INFILE datos;
```

-No pueden encontrarse en un mismo paso DATA alguna sentencia INFILE con sentencias SET, MERGE o UPDATE.

Algunas opciones[8]: (NOTA: registro = línea de datos)

FIRSTOBS = n Primer registro a leer.
OBS = n Número de registros a leer.

Para el caso en que SAS llega al final de un registro de datos según las especificaciones de la sentencia INPUT y según esta sentencia se requieran más datos del mismo registro, pueden especificarse tres opciones:

FLOWOVER (por defecto) se continúa leyendo en la siguiente línea de datos.
MISSOVER asigna valor desconocido a los datos que faltan en la línea.
STOPOVER se para la ejecución del paso DATA.

SENTENCIA INPUT.

La sentencia INPUT:

-Describe la disposición de los datos en entrada y nombra las variables correspondientes.

-Especifica el formato en que se encuentran los datos en entrada: fijo o libre; pudiéndose entremezclar en una misma sentencia INPUT los dos tipos de formato.

-Por defecto, SAS considera que todas las variables son numéricas salvo que se lean con un formato carácter o se especifique el símbolo $ después de su nombre[9].

-En un mismo paso DATA, pueden existir una o más sentencias INPUT para cada fichero INFILE. Una sentencia INPUT hace referencia al último INFILE especificado.

-Cada sentencia INPUT en el paso DATA señala un salto de línea en los datos, salvo indicación contraria.

Posición:

---

[8] Otras opciones son:
RECFM = f Formato de registro de entrada.
LRECL = n Longitud del registro de entrada.
BLKSIZE = n Tamaño de bloque del registro de entrada.
LS = n Longitud de registro que esté accesible por el puntero.
N = n Número de registros de entrada accesibles por el puntero en una misma ejecución del paso DATA. Por defecto es N = max (1, { n / #n aparece en INPUT}).
END = var Se crea la variable lógica var, que será verdadero si se está en la última observación. No se añade al SAS-FILE.

[9] o bien se lo notifique una sentencia ATTRIB anterior.

-Debe ir detrás de alguna sentencia INFILE, salvo que los datos se encuentren en el programa SAS.

Sintaxis:

INPUT especificación [especificación ...] ;

donde especificación puede ser:

-nombre de variable.
-especificación de formato: por columnas, formato SAS de entrada, formato creado por el usuario.
-control de puntero: salto de columnas, salto de línea, posicionamiento en columna o línea.
-Alguno de los siguientes caracteres: $ & ( ) -

Especificación de formato fijo en entrada de datos:

-En este caso, para las variables correspondientes, el formato de entrada es común a todos los casos, ocupando los datos posiciones fijas en determinadas columnas del registro de entrada de datos. El formato fijo se puede especificar por:

-Indicación de columnas. Los valores de las variables correspondientes se localizan en las mismas columnas para todos los casos.

Sintaxis: coli [- colf]

Ejemplo:

```
INPUT  nombre  $  1-8  edad  11-14 ;
```

-Formateados según algún formato SAS.

Sintaxis: formato

Formatos SAS más comunes son:

$charw. o $w. Indica una variable carácter de w caracteres.

w.[d] Indica una variable numérica de w caracteres, d de ellos decimales.

Ejemplo:

```
INPUT  nombre  $char8.  edad  2. ;
```

-[10]

[10] Formateados según un formato propio creado previamente en un PROC FORMAT.
Sintaxis: informato.

Especificación de formato libre en entrada de datos:

-Los valores de las variables para un caso deben estar separados por espacios en blanco, no siendo necesario que se posicionen en columnas fijas.

Ejemplo:

```
INPUT  nombre  $  edad ;
```

Opciones[11] en las especificaciones de formato:

- $ Indica que la variable predecesora es carácter.
- & En formato de lectura libre, permite que la variable carácter predecesora de este símbolo tenga como valor válido de la misma un espacio en blanco entre dos caracteres. La separación con la siguiente variable se realiza por al menos dos espacios en blanco.
- ( ) Agrupa varias variables en las que se repite un formato. Encierra un(os) formato(s) que se repite(n).
- \- Indica una secuenciación numérica de variables. (Ej: x1-x3 indica x1 x2 x3).
- / Indica un salto de línea.
- @n Indica el posicionamiento en la columna n.
- +n Indica el avance de n columnas.
- @ Mantiene el puntero de lectura en un mismo registro de una sentencia INPUT a otra sentencia INPUT del mismo paso DATA y referenciados al mismo INFILE.

Ejemplo:

El valor de la variable tipo indica si un registro de datos contiene información acerca de un profesor o acerca de un alumno.

```
DATA ;
INPUT  tipo  $1.  @ ;
IF TIPO = 'c'  THEN INPUT curso profesor $;
ELSE  INPUT  nombre  $  edad ;
```

- @@ Mantiene el puntero de lectura en la misma línea de datos para la próxima ejecución del paso DATA.

Ejemplo:

Cada línea de datos contiene varios valores de la variable x.

```
DATA ;
INPUT  x  @@ ;
CARDS;
77 72 87 65 63 66 63
```

---

[11] Otras opciones son:

#n Indica el posicionamiento en el registro n-ésimo, en la actual ejecución del paso DATA. La sentencia INFILE con la opción N = k da el rango permitido de movimiento en líneas del puntero de lectura, en una misma ejecución de paso DATA.

```
66 67 69 59 72 67 62
……
;
```

NOTA: Se puede especificar en la sentencia INPUT lectura en formato fijo y libre simultáneamente.

Ejemplo:

```
INPUT nombre $char20. edad 3. +3 sexo 2. @54 estatus 2.
( altura x1-x13 ) ( 5.2 ) / nota $ 16-34 puntos_a
puntos_b ;
```

Entrada de variables de tipo carácter con la sentencia INPUT.

La longitud que se toma para las variables de tipo carácter es de 8. Si se desea incrementar la longitud se debe realizar con la sentencia length antes de la sentencia input.

Ejemplo

```
data a;
length nombre $ 10;
input nombre $ edad;
cards;
MariaElena 56
alicia 22
carmen 23
margarita 34
;
proc print; run;
```

Obs	nombre	edad
1	MariaElena	56
2	alicia	22
3	carmen	23
4	margarita	34

o bien leerlos en columnas fijas y proporcionar en el formato de entrada de datos la longitud de la columna. En el siguiente ejemplo la salida es la misma, salvo en el orden de las observaciones.

Ejemplo:

```
data a;
input nombre $char10. edad;
cards;
alicia     22
carmen     23
margarita  34
MariaElena 56
;
proc print; run;
```

SENTENCIA CARDS.

La sentencia CARDS especifica el comienzo de los datos. Una vez finalizadas las líneas de datos, debe especificarse en una nueva línea la sentencia nula, es decir, un punto y coma (;). La sentencia nula debe ir en la primera columna.

Sintaxis: `CARDS ;`

ESPECIFICACIÓN DE VALORES DE NO OBSERVACIÓN EN LOS DATOS

-Por defecto, para variables numéricas se deberá especificar en los datos el punto (.) para indicar un valor no observado y para variables carácter el blanco ( ). En el caso de valores desconocidos para variables carácter y si éstos son especificados por blancos, la lectura de los datos no puede realizarse mediante formato libre, al menos en las columnas correspondientes a dichas variables carácter.

- En el caso de variables numéricas y para todas las variables simultáneamente puede ampliarse el rango de valores de no observación a valores alfabéticos y _, con la sentencia MISSING.

-Si los valores de no observación especificados en los datos fueran distintos a éstos, mediante programación en el paso DATA (Ej: mediante la sentencia IF) se asignarían los valores de no observación correspondientes: punto (.) para variables numéricas y blanco (' ') para variables carácter.

-SAS asigna valores de no observación cuando hay errores en los datos, se realizan operaciones con valores de no observación, se intentan realizar operaciones fuera de rango, etc. En todos los casos, SAS lo notifica en la salida de diagnósticos.

II.B. ENTRADA DE DATOS ASCII con la INTEFAZ GRÁFICA de SAS

Se muestran tres ficheros de ejemplo que ayudan a la comprensión de este apartado.

Fichero **ejer3.datos**

```
id sexo fuma edad pulso1 pulso2
01 1 1 31 62 126
…………….
09 2 1 22 88 160
10 1 1 28 90 144
```

El conjunto de datos **ejer3.datos** se puede leer con:
Standard Data Source → Delimited
o con *User-defined formats → En opciones: Lista*
y Creación de la variable Automática (variables en la primera fila).
ATENCIÓN: El blanco inicial del primer campo se sustituye por 0.

Fichero **ejer.datos**

```
 1 1 1 31 62 126
     ………….
 9 2 1 22 88 160
10 1 1 28 90 144
```

El conjunto de datos **ejer.datos** se puede leer con *User-defined formats* → *En opciones: Columna y Creación de la variable automática*. Luego, seleccionar los campos y renombrar las variables

Fichero **ejer2.datos**

```
id sexo fuma edad pulso1 pulso2
 1 1 1 31 62 126
…………….
 9 2 1 22 88 160
10 1 1 28 90 144
```

El conjunto de datos **ejer2.datos** se puede leer con con *User-defined formats* → *En opciones: Columna y Creación de variable manual y 2 como Registro de inicio.*

Opción de menú *Archivo/Importar datos*

- Si los datos se encuentran delimitados en columnas fijas (separadas o no por algún carácter: blanco, tabulador, …) **se recomienda** seleccionar la opción *User-defined formats*.
- Si los datos no se encuentran deliminados en columnas fijas y están delimitados por algún carácter[12] (ejemplo: blanco o tabulador), se **puede seleccionar** la opción *Standard Data Source* y seleccionar en source data la opción *fichero delimitado.*

  NOTA IMPORTANTE: *Cuando los datos no están en columnas fijas esta opción puede ser menos sencilla que la lectura con el paso DATA ya que puede no leer correctamente los datos.*
  Por ejemplo:
  -Si el separador de delimitación es un blanco y alguna línea comienza con un blanco, SAS no determina bien el campo. Una posible solución a esto es introducir un 0 en la primera columna de cada línea que comienza con blanco o llevar a la izquierda toda la línea de datos.
  -Tambien en el caso de que el delimitador sea un blanco, si las líneas terminan en blanco, determina que hay variables a la derecha de los últimos datos y les asigna el valor missing.

- Ambas opciones pueden leer en la primera fila el nombre de las variables. Atención: En el primer caso, hay que poner atención a que los nombres de variables se encuentre en las mismas columnas que sus datos.

Puede generar código SAS. Una vez finalizada la lectura de los datos, se puede recuperar el código de SAS en la ventana editor avanzado (o program editor), sin más que dar a la tecla F4 teniendo activa la ventana. Además, la opción Standard Data Source → fichero delimitado permite la creación de código SAS en un fichero independiente.

Archivo/Importar datos -→ Standard data source + Fichero delimitado (Fichero **ejer3.datos**)

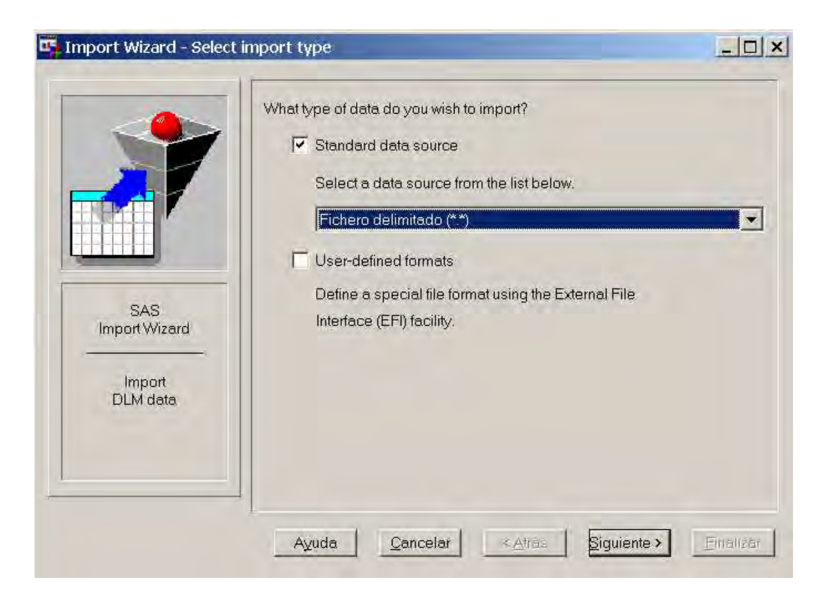
Fichero **ejer3.datos**

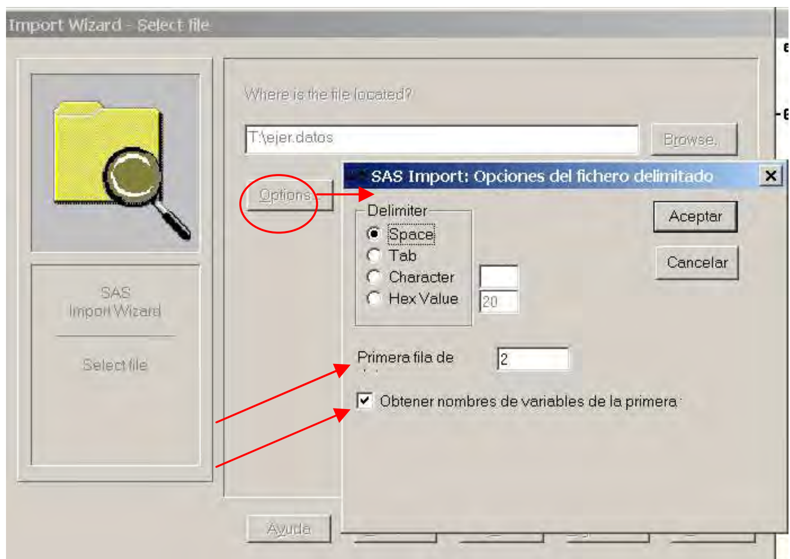
```
id sexo fuma edad pulso1 pulso2
```

[12] Esta opción también es válida para datos que se encuentran deliminados en columnas fijas y están delimitados por algún carácter: blanco, tabulador, …

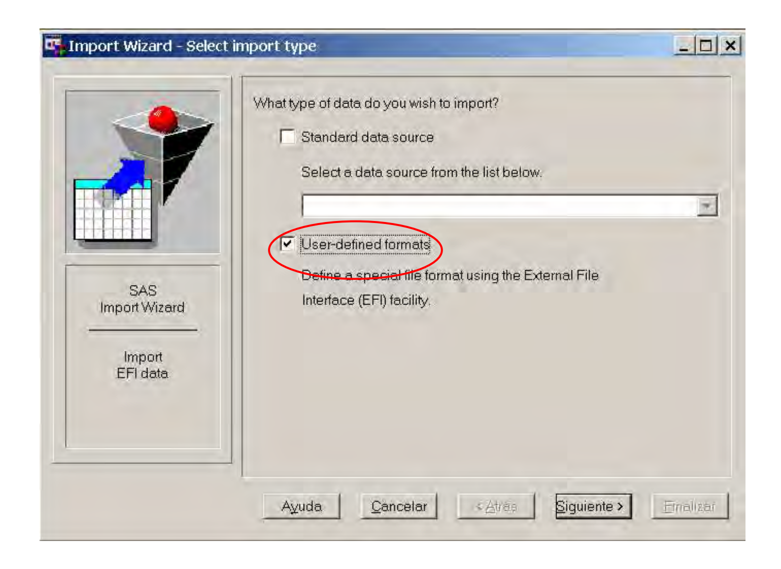
```
01 1 1 31 62 126
…………….
09 2 1 22 88 160
10 1 1 28 90 144
```

ATENCIÓN: El blanco inicial del primer campo se sustituye por 0 (léase nota importante de la página 38).





<u>Archivo/Importar datos -→ User-defined formats</u>



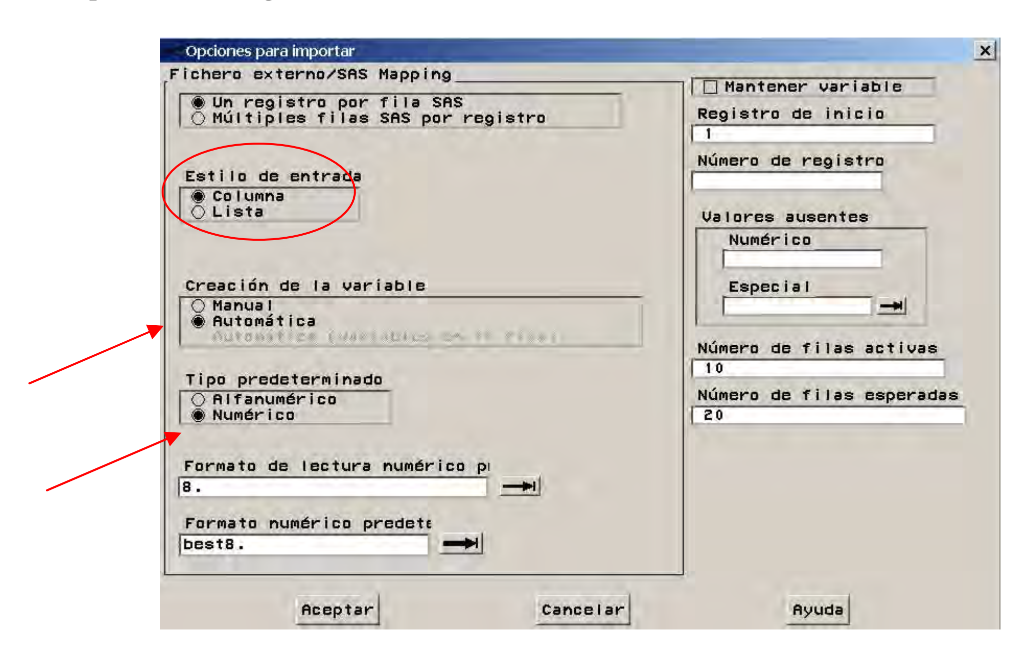
Se van siguiendo los pasos correspondientes en las ventanas sucesivas. Al llegar a la ventana Import, se selecciona el botón *Opciones…* y en el conjunto de opciones se selecciona:

*<u>a) En el caso de que no se encuentren los nombres de variable en el fichero de datos o que no se lean de este fichero (Fichero **ejer.datos**):</u>*

Fichero **ejer.datos**

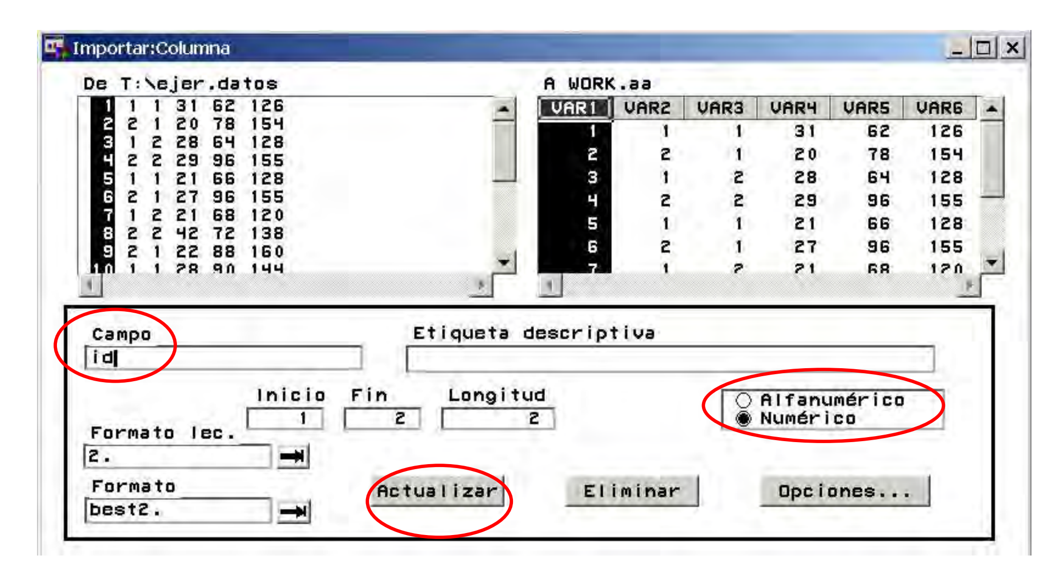
```
 1 1 1 31 62 126
   ..........
 9 2 1 22 88 160
10 1 1 28 90 144
```

Estilo de entrada → Columna.
Si se desea que el tipo de las variables por defecto sean numéricas seleccionaremos la opción Tipo predeterminado → numérico.
Además si deseamos que cree las variables automáticamente, seleccionaremos la opción Creación de la variable → Automática
Si los nombres se encuentran en este fichero, en la primera línea de datos, debe especificarse registro de inicio: 2.



Seleccionadas las opciones correspondientes se pulsa *aceptar* y en la ventana de importación de datos a la que se vuelve, se puede cambiar para cada variable la opción de tipo de variable y se pueden nombrar (o renombrar) las variables o borrarlas.

Las opciones seleccionadas en el panel anterior nos devuelven nos devuelven el siguiente aspecto del panel de importación de datos:



donde para nombrar las variables, basta con seleccionarlas y darles un nombre y pulsar el botón *actualizar*. Si deseamos borrar una variable, la seleccionamos y pulsamos *Eliminar* y si deseamos cambiar el tipo en carácter/numerico, seleccionamos la opción deseada.

*b) En el caso de que se encuentren los nombres de variable en el fichero de datos y se lean de este fichero (fichero **ejer3.datos** y **ejer2.datos** (ver c))):*

Fichero **ejer3.datos**

```
id sexo fuma edad pulso1 pulso2
01 1 1 31 62 126
...............
09 2 1 22 88 160
10 1 1 28 90 144
```

ATENCIÓN: El blanco inicial del primer campo se sustituye por 0 (léase nota importante de la página 38).

Se deben seleccionar las opciones

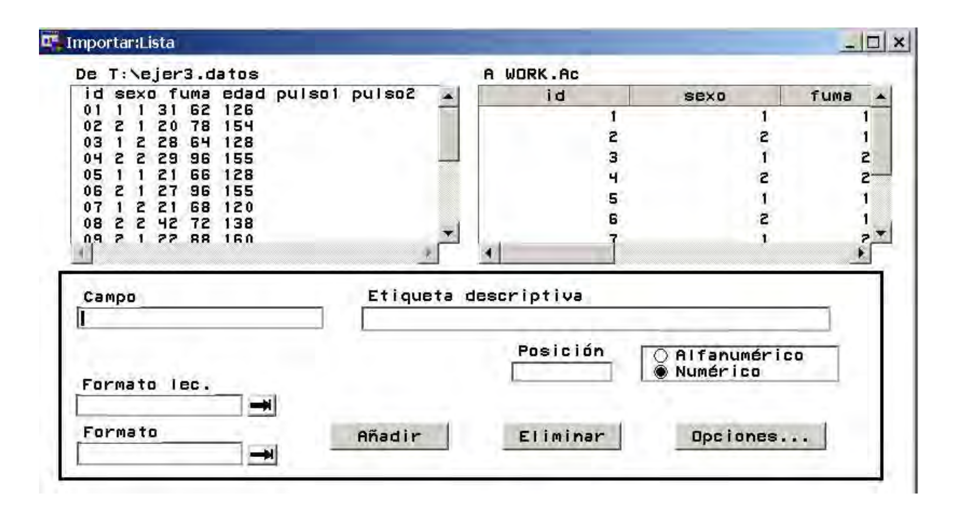
Estilo de entrada → Lista

Si se desea que el tipo de las variables por defecto sean numéricas seleccionaremos la opción Tipo predeterminado → numérico.

Además si deseamos que cree las variables automáticamente y tome el nombre de la primera línea , seleccionaremos la opción Creación de la variable → Automática (variables en 1ª fila)

Seleccionar la opción sep variables y especificar en registro de inicio: 2.

devuelve el siguiente panel de importación de datos.



NOTA IMPORTANTE: *En este modo lista si existen blancos (u otros delimitadores) en las posiciones siguientes: :*

-alguna línea comienza con un delimitador

-líneas terminan en delimitador

entonces SAS no determina bien el campo. Una posible solución a esto es, en el primer caso, introducir un 0 en la primera columna de cada línea de datos que comienza con delimitador o llevar a la izquierda toda la línea de datos y, en el segundo caso, determina que hay variables a la derecha de los últimos datos y les asigna el valor missing (para solucinarlo eliminar estos delimitadores).

Opcionalmente, para solucionar este problema, se puede leer como en a)

*c) Creación manual de variables (Casos a) y b)) (fichero **ejer2.datos**)*

Se puede seleccionar en ambos casos a) y b) que la creación de variables sea manual.

Fichero **ejer2.datos**

```
id sexo fuma edad pulso1 pulso2
 1 1 1 31 62 126
…………….
```

```
 9 2 1 22 88 160
10 1 1 28 90 144
```

Se deben seleccionar las opciones

Estilo de entrada → Columna

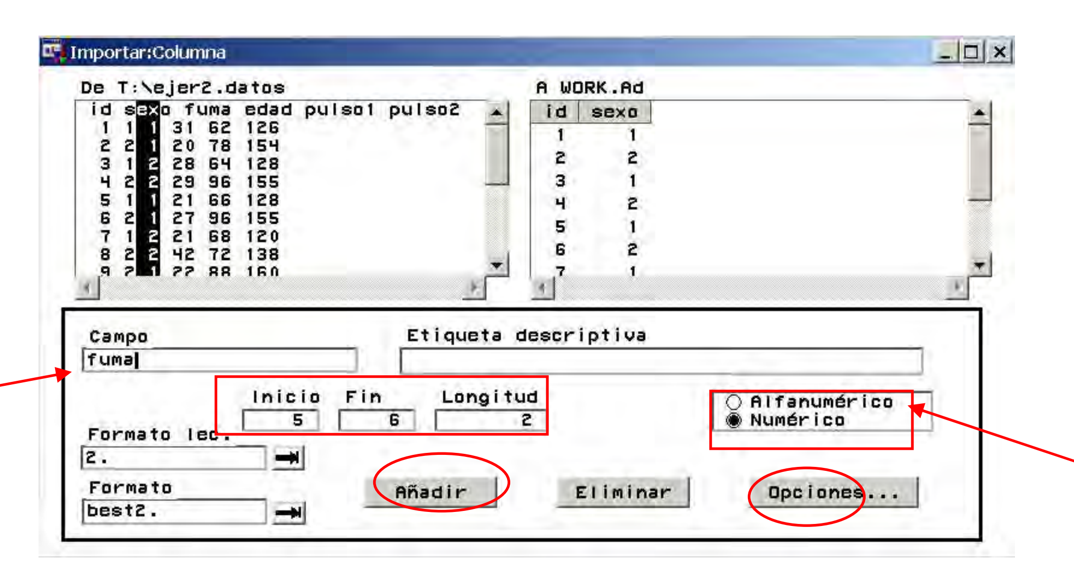
Si se desea que el tipo de las variables por defecto sean numéricas seleccionaremos la opción Tipo predeterminado → numérico.

Creación de la variable → Manual

Especificar en registro de inicio: 2.

Al volver al panel de los datos, se deben incluir en el panel de la derecha que muestra el SAS-FILE los datos de variable en variable, indicando el tipo de variable y las columnas de inicio y final de la misma. Si la columna se selecciona con el ratón, esta información se incorpora automáticamente en el panel. Se da un nombre a la variable y se indica si es de tipo carácter o numerico.

Finalmente, esta columna se incorpora al SAS-FILE con la opción *añadir*.



De modo similar se puede hacer para el mismo fichero cuando están ausentes las nombres de las variables (sin embargo nos parece más cómodo hacerlo como en a)).

Fichero **ejer.datos**

```
 1 1 1 31 62 126
...............
 9 2 1 22 88 160
10 1 1 28 90 144
```

El único cambio con respecto a la página anterior es que se especifica que el registro de inicio es el primero. Es decir, se deben seleccionar las opciones

Estilo de entrada → Columna

Si se desea que el tipo de las variables por defecto sean numéricas seleccionaremos la opción Tipo predeterminado → numérico.

Creación de la variable → Manual

Especificar en registro de inicio: **1**.

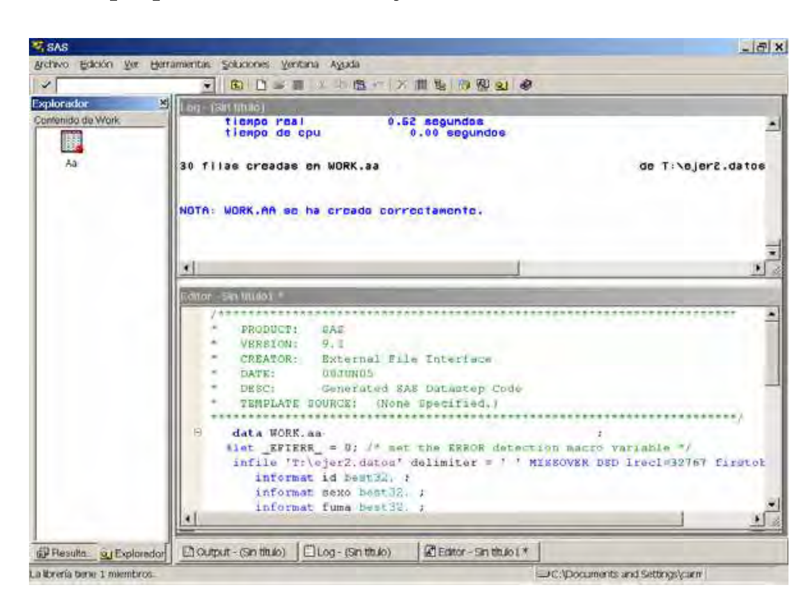
Al volver al panel de los datos, se deben incluir en el panel de la derecha que muestra el SAS-FILE los datos de variable en variable, indicando el tipo de variable y las columnas de inicio y final de la misma. Si la columna se selecciona con el ratón, esta información se incorpora automáticamente en el panel. Se da un nombre a la variable y se indica si es de tipo carácter o numerico.

Finalmente, esta columna se incorpora al SAS-FILE con la opción *añadir*.

<u>EN TODOS LOS CASOS:</u>

Para terminar la creación del SAS-FILE, una vez creadas todas las columnas, se accede a la opción de menu *Archivo/Guardar* y se proporciona un nombre al mismo.

Código SAS que se genera al pulsar la tecla F4 cuando la ventana activa es *editor avanzado*. (En este caso el nombre proporcionado es AA y solo contiene una columna de nombre ID :



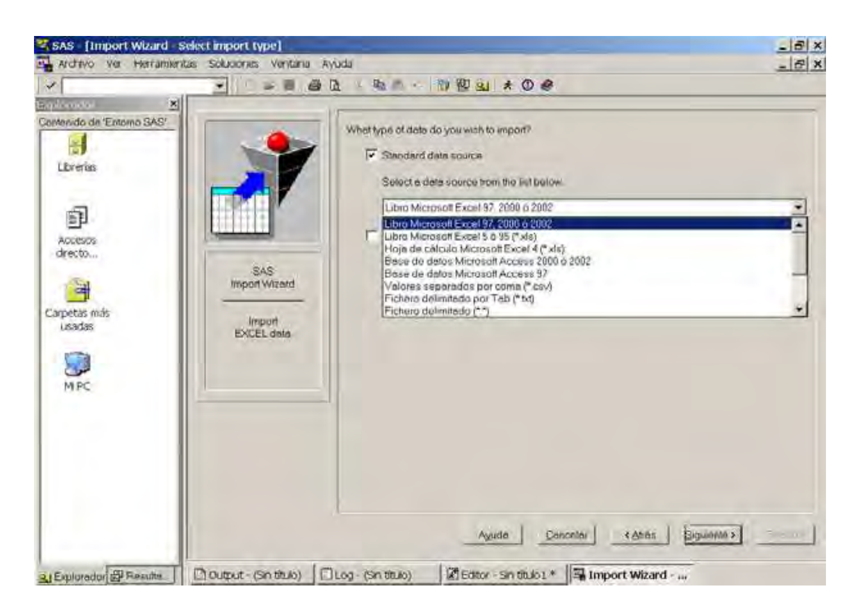
<u>II.C. IMPORTACIÓN (EXPORTACIÓN) DE DATOS EN OTROS FORMATOS: EXCELL,...</u>

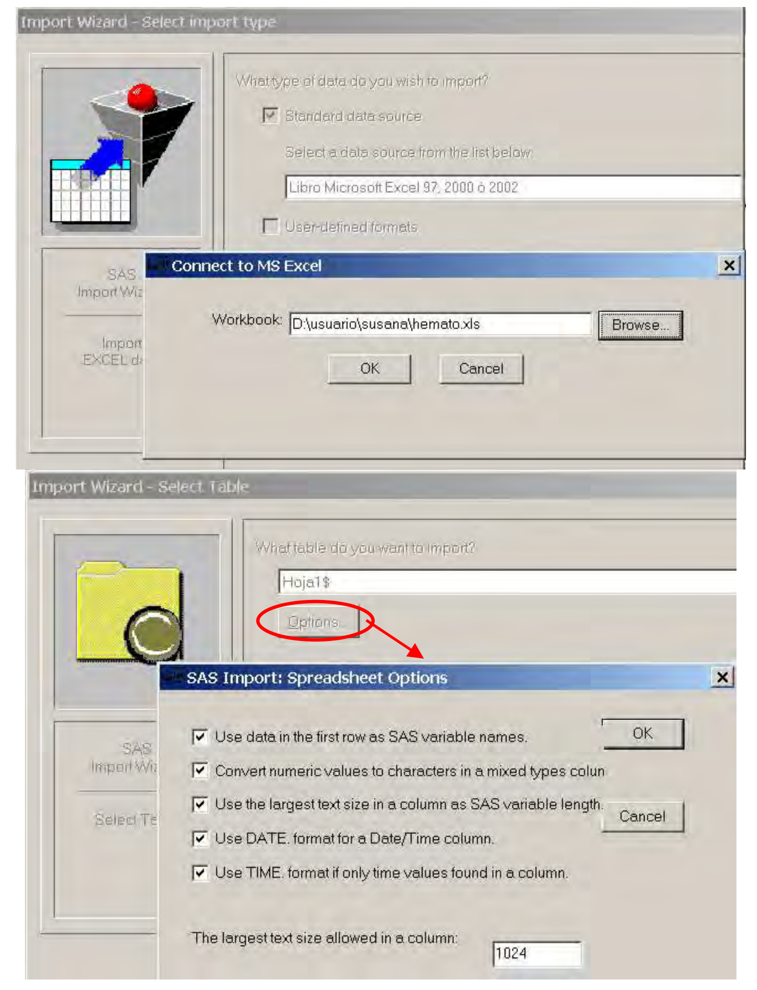
Se presenta aquí también la exportación de datos ya que es similar a la importación de los mismos.

Se puede realizar con los procedimientos Import o Export. Aquí mostramos cómo realizarlo con el entorno gráfico de SAS. Para ello:

- Seleccionar en el menu **archivo** las opciones de **importar datos o exportar datos**
- Seleccionar el tipo de formato externo de datos y pulsar siguiente
- Indicar el fichero de datos de lectura
- Indicar la hoja que se desea importar y activar Options para seleccionar las opciones correspondientes
- Seleccionar el nombre del SAS-FILE que se desea crear
- Opcionalmente, indicar un nombre de fichero para almacenar el código SAS correspondiente

En las siguientes imágenes se muestra la secuencia de ventanas correspondientes para la importación de datos de un fichero Excel.

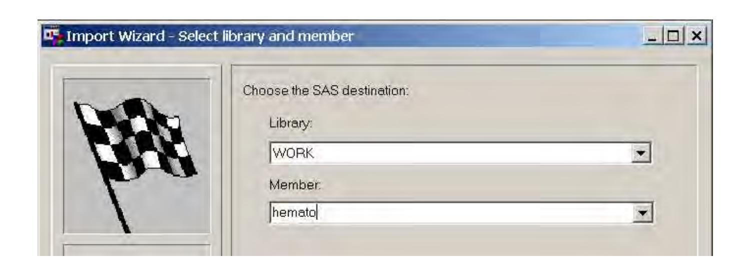




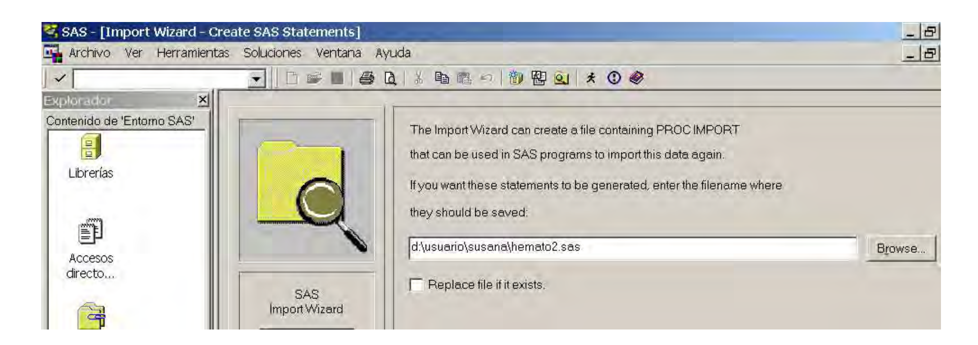
Si el formato de entrada de datos seleccionado es excell 95 la última ventana es reemplazada por:



Seguimos con los paneles de importación de datos



- Opcionalmente se puede crear el código SAS que genera la importación/exportación de datos.



En este caso el código generado se encuentra en el fichero d:\usuario\susana\hemato2.sas y contiene:

```
PROC IMPORT OUT= WORK.hemato
            DATAFILE= "D:\usuario\susana\hemato.xls"
            DBMS=EXCEL REPLACE;
     SHEET="Hoja1$";
     GETNAMES=YES;
     MIXED=YES;
     SCANTEXT=YES;
     USEDATE=YES;
     SCANTIME=YES;

PROC PRINT DATA=hemato;

RUN;
```

Se puede recuperar este código SAS[13] generado abriendo el fichero de sentencias SAS desde la ventana editor avanzado (o program editor) con la opción archivo → abrir programa, recupera el código SAS.

ATENCION: Importar ficheros Excel en SAS puede generar datos desconocidos (MISSING)

Este problema (los datos de algunas columnas pueden importarse como missing) puede suceder con la importación de ficheros Excel 97 o superiores si hay datos de tipos mezclados en la misma columna. SAS determinará el tipo apropiado para los datos basándose en el más común para las ocho primeras filas

Existen dos[14] soluciones para este problema:

Solución 1:

- En la importación de datos de excell, seleccionar la opción convert numeric values to carácter values in a mixed type column. Esto hace que las columnas que tienen alguna celda en blanco, sean leídas como de tipo carácter.
- Crear posteriormente una variable de distinto nombre que sea la variable carácter leída más el valor 0.

Ejemplo: Código producido por la interfaz gráfico y paso DATA que resuelve el problema descrito, para la variable pulso1.

```
PROC IMPORT OUT= WORK.PRIM
            DATAFILE= "T:\EJERB1.XLS"
            DBMS=EXCEL REPLACE;
     SHEET="ejer#datos$";
     GETNAMES=YES;
     MIXED=YES;
     SCANTEXT=YES;
     USEDATE=YES;
     SCANTIME=YES;

DATA PRIM;
SET PRIM;
PULSO1B=PULSO1+0;
DROP PULSO1;
```

Solución 2 (**recomendada**):

- Grabar el fichero excel en formato excel5 o excel95
- Realizar la importación de los datos, seleccionando este formato.
- Revisar que los datos leídos son correctos. (Las pruebas realizadas por mí han sido satisfactorias)

II.D. INTRODUCCIÓN de DATOS en SAS® con la INTERFAZ GRÁFICA

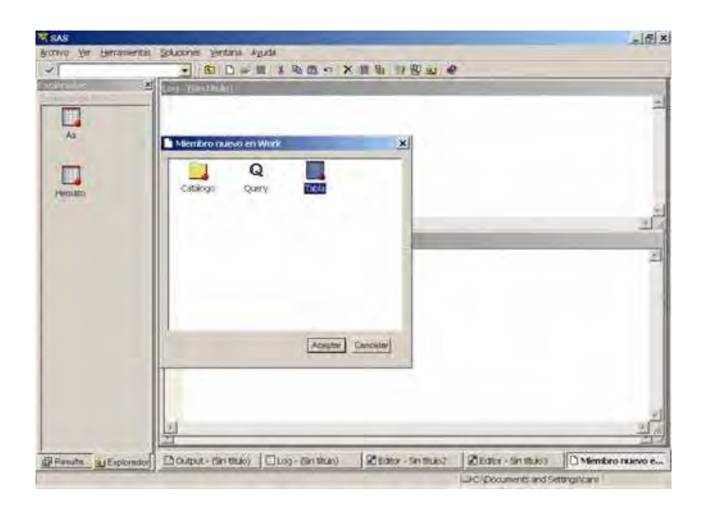
---

[13] **La tecla F4 NO facilita en la ventana *editor* el código relativo a la importación y exportación de datos con el entorno gráfico**

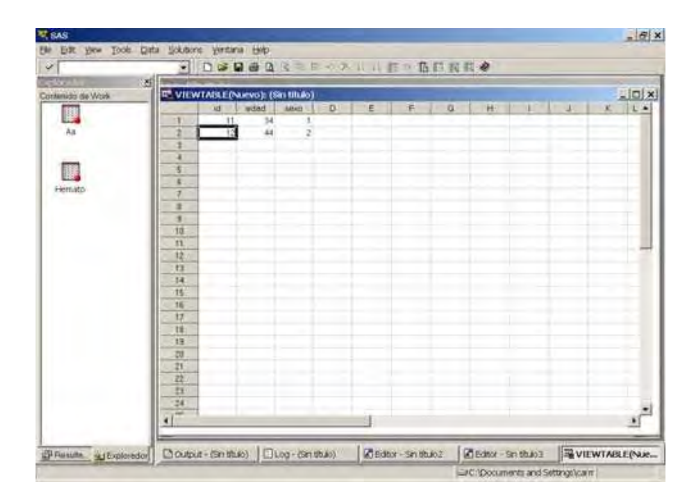
[14] **Existen otras soluciones alternativas. Comprobar el resultado de las dos soluciónes presentadas, posteriormente a su realización.**

Por lo general, los datos se encontrarán en un fichero externo en ASCII u otro formato (EXCEL, dbase, etc…). Aún así, mostramos cómo podrían ser introducidos los datos en SAS con la interfaz gráfico para crear SAS-files.

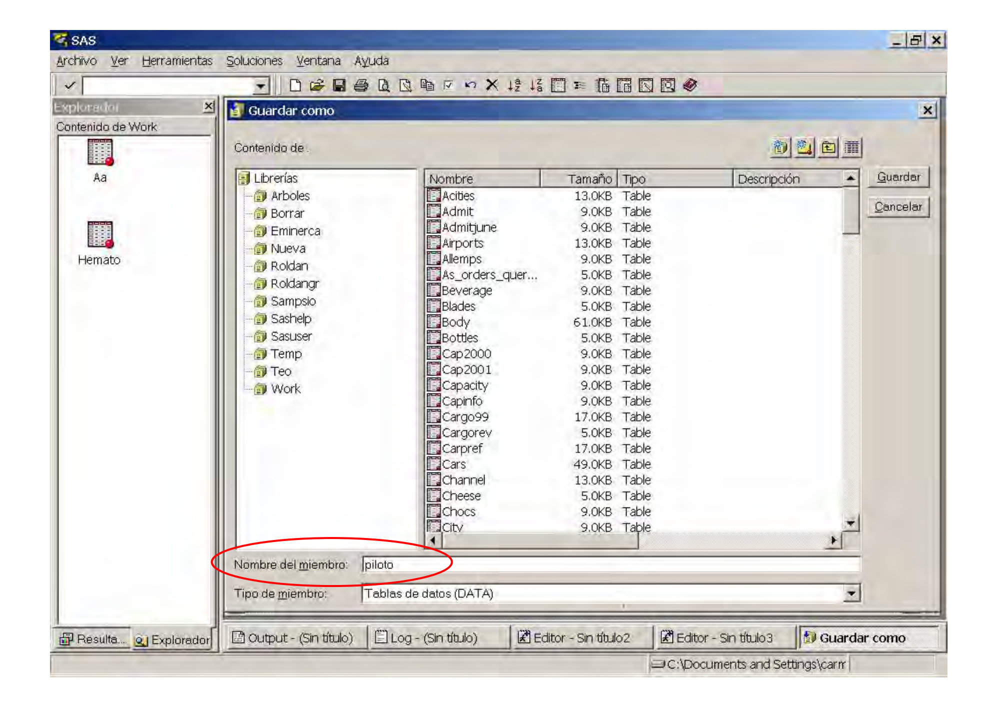
Activada la ventana *explorador* cuando se esta en los contenidos de una librería o seleccionada con el ratón una librería: *Archivo/Nuevo/Tabla*:



Se introducen los nombres de variable y los datos, posicionando el ratón en las celdas correspondientes:



Antes de salir se nombre el SAS-File (*File/Save as* o SAS pregunta al desear salir). En el panel que aparece para nombrar y almacenar el SAS-FILE, se debe abrir con doble click de ratón izquierdo, la librería en la que se desea almacenar el SAS-FILE.



Esta utilidad no permite recuperar código SAS.

III. PASO DATA. UTILIDADES.

III.1 TRANSFORMACIÓN O CREACIÓN DE NUEVAS VARIABLES

Sentencias de asignación.

La transformación o creación de nuevas variables se realiza mediante sentencias de asignación del tipo:

```
A = B ;
```

siendo A el nombre de una variable y B una expresión aritmética o carácter.

Las expresiones aritméticas son de la forma:

```
va1  [ operador-aritmético  va2  ... ]
```

donde vai puede ser:

-variable
-constante
-expresión aritmética
-función aplicada a una variable o constante

Operadores aritméticos son:

+ : adición
- : sustracción
* : multiplicación
/ : división
** : exponenciación

-Se puede hacer uso de paréntesis dentro de la expresión.
-En la expresión el orden de precedencia viene dado por: ( ), **, / y * indistintamente aquél que aparezca primero en la expresión; y - y + indistintamente también.

Ejemplo:

```
A * 7 / 4 - X * Y**2 + 3  equivale a  (( A * 7 ) / 4 ) - ( X * ( Y**2 )) + 3
```

Funciones SAS.

-Se usan en los pasos DATA como expresiones o parte de las mismas.
-Son de muy diversas utilidades:
Aritméticas: ABS, MAX, MIN, SQRT ...
Carácter.
Financieras.
Fecha y hora.
Matemáticas: GAMMA, LOG, LOG2, LOG10, EXP, ERF ...
Probabilísticas y estadísticas: POISSON, PROBBETA, PROGCHI, PROBNORM, PROBT, NORMAL, RANBIN, MEAN, SKEWNESS, PROBIT, FINV ...
Trigonométricas: ARCOS, SIN, TANH ...
De truncación.
Especiales.

Sintaxis:

nombre-función (argumento, ...)
o CALL nombre-función (argumento, ...)

Expresiones con valores no observados.

Si en una expresión o en una función alguno de los términos o argumentos, respectivamente, tiene valor desconocido, el resultado de las mismas es desconocido.

Ejemplo:

```
logX = LOG(X) ;
Xmean = MEAN (X1,X2,X3,X4);
z = x**2 + y**2;
```

III.2 UTILIDADES EN EL PASO DATA

Renombrar variables.

-Se realiza mediante la sentencia RENAME.

-RENAME es también una opción de SAS-FILE.

Sintaxis:

RENAME nombre-viejo = nombre nuevo ... ;

-no se permite referenciar listas de variables en ninguno de los dos términos de la expresión.

Selección de variables en un SAS-FILE.

-Se realiza mediante las sentencias KEEP o DROP.
-La sentencia KEEP selecciona la variables a mantener en un SAS-FILE.
-La sentencia DROP selecciona las variables a eliminar de un SAS-FILE.
-Estas dos sentencias se pueden repetir en un mismo paso DATA, pero nunca deben usarse simultáneamente.
-KEEP y DROP son también opciones de SAS-FILE.

Sintaxis:

KEEP | DROP variable ... ;

Ejemplo:

Estos tres ejemplos son equivalentes:

```
DATA ejemplo;
INFILE 'ejemplo.dat';
INPUT pac sexo peso altura hdl ldl htamin htamax;
bmi = peso**2 / altura ;
drop peso altura;

DATA ejemplo;
INFILE 'ejemplo.dat';
INPUT pac sexo peso altura hdl ldl htamin htamax;
bmi = peso**2 / altura ;
keep pac sexo hdl ldl htamin htamax bmi;

DATA ejemplo;
INFILE 'ejemplo.dat';
INPUT pac sexo peso altura hdl ldl htamin htamax;
bmi = peso**2 / altura ;
keep pac sexo hdl--bmi;
```

Selección de observaciones en un SAS-FILE.

-Se realiza mediante las sentencias DELETE o OUTPUT.
-La sentencia DELETE selecciona las observaciones a eliminar de un SAS-FILE.
-La sentencia OUTPUT selecciona las observaciones a incluir en un SAS-FILE.
-Las sentencias IF o WHERE pueden utilizarse para la selección de observaciones.
-La sentencia IF se diferencia de la sentencia WHERE, en que en la primera se leen todos los valores de observación, y en la segunda se leen primero los implicados en la sentencia WHERE, y si la expresión da valor verdadero, se leen los demás.
-WHERE es también una opción de SAS-FILE.

Sintaxis:

```
DELETE ;
OUTPUT  [ SAS-file ... ] ;
IF  |  WHERE  condición-lógica ;
```

(NOTA: Véase el bloque IF / THEN / ELSE para la especificación condición-lógica)

Ejemplo:

Son equivalentes:

```
DATA ;
INPUT  x  y ;
IF  x < 5  THEN  OUTPUT ;
CARDS ;

DATA ;
INPUT  x  y ;
IF  x < 5 ;
CARDS ;

DATA ;
INPUT  x  y ;
IF  x >= 5  THEN  DELETE ;
CARDS ;
```

Creación de diferentes SAS-FILEs en un mismo paso DATA.

-Se puede realizar mediante la sentencia OUTPUT.

-Mediante las opciones de SAS-FILE KEEP y/o DROP, se pueden seleccionar las variables a incluir en cada uno de los SAS-FILEs creados.

Ejemplo:

```
DATA  nino  joven  ;
INFILE  'mis.datos' ;
INPUT  nombre  $  sexo  $  edad ;
IF  edad <= 14  THEN  OUTPUT  nino ;
ELSE  OUTPUT  joven ;
```

Ejemplo:

```
DATA persona(KEEP = id nombre dir) estad(DROP = nombre dir);
INFILE  ‘c:\censo.dat’ ;
INPUT  id  nombre  $char20.  dir  $char20.  x1-x20 ;
```

Creación de más de una observación por línea de datos.

-Se puede realizar mediante la sentencia OUTPUT.

Ejemplo:

```
DATA ;
INPUT  id  x1-x3 ;
DROP   x1-x3 ;
x = x1 ;  OUTPUT ;
x = x2 ;  OUTPUT ;
x = x3 ;  OUTPUT ;
CARDS ;
```

Codificación de variables.

-Puede asignarse un descodificado a las categorías de una variable cualitativa, mediante la sentencia FORMAT, si previamente se ha creado un formato adecuado en el procedimiento FORMAT.

-Las variables continuas se pueden categorizar mediante un formato creado en PROC FORMAT, que haga asignaciones de categorías por intervalos. De esta forma, las variables a las que se asigna un formato con categorías, pueden ser consideradas categóricas (si se usan en una sentencia BY, CLASS, TABLE, TABLES) o continuas (si se usan en una sentencia VAR o no se hace mención explícita de ellas) puesto que el valor numérico original de las mismas se conserva.

-La sentencia FORMAT asocia formatos, creados en un PROC FORMAT, a variables.

N.B. Véase el procedimiento FORMAT más adelante de esta guía.

Sintaxis:

FORMAT   especificacion-variables nombre-formato. ... ;

Ejemplo:

FORMAT edad edad. x1-x10 sino.;

III.3.A. (AVANZADO) CREACIÓN DE FICHEROS EXTERNOS DE DATOS O INFORMES

El paso DATA puede ser de utilidad para la salida de ficheros de datos, bien por la facilidad de programación en SAS, bien por la salida de resultados interesantes de procedimientos SAS.

Asimismo, el paso DATA puede usarse para la generación de informes.

Ejemplo básico:

```
DATA  _NULL_;
INFILE  'morfo.datos';
INPUT  id  $4.  (tamaño  altura  peso) (6.2)  sexo 2.;
FILE  'morfosal.datos';
IF  (sexo = 9)  THEN  sexo = . ;
bmi = (peso**2) / (altura*100);
PUT  id  $4.  (tamaño  altura  peso) (6.2)  sexo 2.  bmi 7.2;
```

SENTENCIA FILE

La sentencia FILE:

-Identifica como salida, un fichero externo de datos en forma decimal. La identificación puede ser mediante el nombre del fichero en disco, encerrado por comillas ('), o bien mediante una referencia o nombre lógico que identifica el fichero. Cuando se especifica el nombre de un fichero es conveniente especificar la ruta completa del fichero.

-En el caso de especificación de una referencia, debe existir antes del paso DATA una sentencia FILENAME. La sentencia FILENAME se describe en el apartado II.3, sentencia INFILE.

-Describe las características del fichero externo.

-Especifica cabeceras que deben escribirse al principio de cada página del fichero, si el fichero es de tipo PRINT, es decir, si la referencia que identifica el fichero es PRINT o se especifica la opción PRINT.

-Indica modo de actuación cuando se pretende escribir mas allá de la longitud del fichero.

Sintaxis:

FILE 'nombre-fichero' | ref [opcion ...] ;

Existen algunas referencias especiales que pueden especificarse en la sentencia FILE:

-PRINT	indica el dispositivo de salida de resultados.
-LOG	indica el dispositivo de salida de diagnósticos.

Posición:

-Debe ir precediendo la sentencia PUT correspondiente.

Observación:

-Puede haber más de una sentencia FILE si se escriben datos en varios ficheros externos. Una sentencia PUT, en este caso, escribirá en el fichero reseñado en el último FILE.

Algunas opciones:

RECFM = f	Formato de registro del fichero de salida.
LRECL = n	Longitud de registro del fichero de salida.
BLKSIZE = n	Tamaño de bloque del registro del fichero de salida.
PRINT	Hace el fichero de tipo PRINT, es decir, pone control de carro. No es necesario si se especificó la referencia PRINT.
NOPRINT	No pone control de carro cuando se especifica la referencia de PRINT.

HEADER = etiqueta	Define la etiqueta de la sentencia que encabeza el grupo de sentencias que se procesan a cada nueva página. El final de dichas sentencias viene dado por RETURN; (Ficheros PRINT).
N = n | PS	Número de líneas del fichero de salida, accesibles por el puntero en una iteración del paso DATA. Si PS, será toda la página. Si no se especifica, será N = max (1, n de #n).
PS = n	Número de líneas por página. ([20, 500]). Se incluyen las líneas ocupadas por los TITLE. (Ficheros PRINT).
MOD | OLD (por defecto)	MOD indica que se escriba a continuación del fichero, OLD indica que se cree por primera vez.

Opciones al acabar una línea del fichero de salida, según las especificaciones dadas en la sentencia PUT:

FLOWOVER (por defecto) se continúa escribiendo en la siguiente línea.
DROPOVER no escribe los datos que excedan la longitud de línea.
STOPOVER se para la ejecución del paso DATA.

SENTENCIA PUT.

La sentencia PUT:

-Describe la disposición de los datos en salida y especifica el formato de escritura: fijo o libre; pudiéndose entremezclar en una misma sentencia PUT los dos tipos de formato.

-Pueden escribirse, además, cadenas de caracteres si se encierran entre comillas (') en la sentencia PUT.

-En un mismo paso DATA, pueden existir una o más sentencias PUT para cada fichero FILE. Una sentencia PUT hace referencia al último FILE especificado.

-Puede no especificarse una sentencia FILE. En este caso, se escribirá en la salida de diagnósticos.

-Cada sentencia PUT en el paso DATA señala un salto de línea en la salida, salvo indicación contraria.

Sintaxis:

PUT [especificación ...] ;

donde especificación puede ser:

-Nombre de variable [=] Si se pone el signo = aparecerá en el fichero de salida, el nombre de la variables seguido de = y seguido del valor de la variable. En otro caso, aparecerá únicamente el valor de la variable.
-Especificación de formato: por columnas, formato de salida: SAS o creado por le usuario en un PROC FORMAT.
-Control de puntero: salto de columnas, salto de línea, posicionamiento en columna o línea.

-Alguno de los siguientes caracteres: $ ( ) - En este caso, $ no es necesario para indicar que una variable es carácter.
-'Cadena de caracteres'
-_INFILE_: escribe la última línea leída.
-_ALL_: escribe los valores de todas las variables incluyendo _ERROR_ y _N_. _ERROR_ indica si hay error en la observación; _N_ indica el número de veces que se ha ejecutado el paso DATA.
-_PAGE_: salta de página en el fichero de salida.

Especificación de formatos:

-Se realiza del mismo modo que en la sentencia INPUT.

- $ ( ) - / @n +n #n @ y @@ tienen el mismo significado que en la sentencia INPUT.

Opciones adicionales:

-@var posicionamiento en la columna indicada por la variable var.

-#var posicionamiento en la línea indicada por la variable var. Si no se especifica la opción N = n en la sentencia FILE, por defecto SAS toma el valor del mayor valor que acompaña a un #.

III.3.B. EXPORTACIÓN DE DATOS A OTROS FORMATOS CON EL ENTORNO GRÁFICO DE SAS

Con la opción de menú *Archivo/Exportar datos* se pueden exportar datos a ficheros en formato ASCII de forma similar a como se hace con la entrada de datos en II.B (página 33). (Internamente se realiza con un paso DATA).

Se pueden exportar datos a otros formatos, como EXCEL, de forma similar a como se hace con la importación de datos en II.C. (Internamente se realiza con un proc EXPORT).

III.4 SENTENCIAS LOGICAS Y BUCLES

Bloques IF / THEN / ELSE.

Sintaxis:

IF expresion-logica [THEN accion1 ; [ELSE accion2;]]

Observaciones:

-Si la expresion-logica toma el valor verdadero se ejecutará la acción accion1 y sino se ejecutará la acción accion2.

-Cuando no aparece la cláusula THEN, debe especificarse un punto y coma (;) después de la expresión lógica. Se utiliza para seleccionar observaciones en un SAS-FILE.
-Si no aparece THEN no se puede especificar la sentencia ELSE.
-accion1 y accion2 son sentencias ejecutables, o bloques de sentencias comenzando por la sentencia DO.

Las expresiones lógicas son de la forma:

va1 operador va2 ...

donde vai puede ser:

-variable o función aplicada a una variable o constante
-constante
-expresión aritmética
-expresión lógica

Operadores son:

> ó GT : mayor que	>= ó GE : mayor o igual que
= ó EQ : igual que	¬= ó NE : distinto a
<= o LE : menor o igual que	< ó LT : menor que
& ó AND : y	| ó OR : o
¬ ó NOT : no	

-Se puede hacer uso de paréntesis dentro de la expresión.

Valores de no observación en expresiones lógicas:

-Tienen el valor de falso en combinación con los operadores AND y OR.

Ejemplo:

```
IF ( ( (a + b) > 5 ) OR ( id = 'novale') )  THEN y = a ** 2;
ELSE  y = a ;
```

Bucles[15]. Bloques DO / END.

---

15 Sintaxis completa de los bucles:

DO [indice = n TO m [BY k]] [WHILE | UNTIL (condicion)] ;
(sentencias SAS)
END ;

ó DO OVER nombre-array ;
(sentencias SAS)
END ;

Observaciones distintivas cuando se especifican las sentencias WHILE o UNTIL:
-El grupo de sentencias entre las sentencias DO y END se ejecuta (en una misma ejecución de paso DATA) mientras el valor de la variable índice no supere m o mientras la condición especificada en WHILE o UNTIL es verdadera.

Sintaxis:

DO [indice = n TO m [BY k]];
(sentencias SAS)
END ;

Observaciones:

-Si la sentencia DO aparece sin indice, se ejecutan las sentencias entre DO y END, al llegar a la sentencia DO.
-En otro caso, el grupo de sentencias entre las sentencias DO y END se ejecuta (en una misma ejecución de paso DATA) mientras el valor de la variable índice no supere m.
-n, m y k pueden ser valores numéricos o variables numéricas que tomen algún valor.
-Se puede especificar como valores posibles del índice una lista de valores:
DO indice = val1, ... valn ;
-La variable índice puede ser numérica o carácter.
-Se puede salir de un bucle con una sentencia GOTO o GO TO.
-Se pueden anidar bucles.

Ejemplo:

```
DATA ventas;
INFILE 'fichero.ventas';
INPUT empresa ventas1-ventas4;
ARRAY vent{4} ventas1-ventas4;
ventast = 0;
DO i = 1 TO 4;
ventast = ventast + vent{i};
end;
DROP i;
```

Sentencia GOTO o GO TO.

Sintaxis:

GOTO etiqueta ; equivale a: GO TO etiqueta ;

Hay otra sentencia en el paso DATA que comienza por:

etiqueta: sentencia-SAS ;

La sentencia GO TO o GOTO lleva el flujo de control a la sentencia referenciada por la etiqueta.

Otros bloques de sentencias[16]

-WHILE chequea la condición antes de la ejecución de la primera sentencia del DO.
-UNTIL chequea la condición justo antes de la sentencia END. Por tanto, se ejecuta al menos una vez.
[16] **Bloque SELECT / WHEN / OTHERWISE / END.**

III.5 OTRAS SENTENCIAS USADAS EN EL PASO DATA

Para poder referenciar variables, se pueden construir arrays[17] con la sentencia:

ARRAY nombre-arrray { n1 } [$] especificacion-variables ;

-Todas las variables deben ser del mismo tipo, numérico o carácter. Si son de tipo carácter, debe especificarse el símbolo $.
-Posterior a esta sentencia y dentro del mismo paso DATA, ya que únicamente es una denominación de las variables, para referenciar la variable se puede hacer por el nombre de la variable o por nombre-array{i} siendo i la posición que ocupa la variable en el array.

Ejemplo:

Sintaxis:

```
SELECT [ ( expresion ) ] ;
WHEN (expresion2 ) [ accion2 ] ;
...
OTHERWISE [ accionn ] ;
END ;
```

Observaciones:

-Se evalúa la expresión de la sentencia SELECT y las expresiones de las sentencias WHEN. Sucesivamente se van comparando los valores de la expresión SELECT con los valores de las expresiones. Al coincidir la primera vez los valores, se ejecuta la accion correspondiente y finaliza la ejecución del grupo SELECT, saliendo de él.
-Las acciones pueden ser un bloque DO, SELECT o la sentencia nula (;). En ocasiones, es conveniente poner la sentencia nula en la sentencia OTHERWISE, para que no de error el SELECT si no se cumple ninguna expresión WHEN.

Ejemplo: (Son equivalentes)

```
SELECT (a) ;                             SELECT ;
      WHEN (1) x = x*10 ;                      WHEN (a = 1) x = x*10 ;
      WHEN (2) ;                               WHEN (a = 2) ;
      WHEN (3, 4, 5) x = x*100 ;               WHEN (a=3|a=4|a=5) x = x*100;
      OTHERWISE ;                        OTHERWISE ;
END ;                                    END;
```

Sentencias LINK y RETURN.

Sintaxis:

```
LINK   etiqueta;
```

Hay un bloque en el paso DATA:

```
etiqueta: sentencia-SAS ;
setencia-SAS ; ...
RETURN ;
```

La sentencia LINK lleva el flujo de control a la sentencia referenciada por la etiqueta. Se ejecutan las sentencias SAS que se encuentran hasta una sentencia RETURN; , la cual devuelve el control a la siguiente sentencia posterior al LINK.

[17] La sintaxis completa de la sentencia ARRAY es:

ARRAY nombre-vector { n1 [, n2...] } [$] especificacion-variables ;

-En caso de especificación de más de un índice, el que se mueve más rápido es el último. En dos dimensiones se llenan primero las filas.

```
DATA ejemplo;
SET ejemplo;
ARRAY todas{20} edad -- glucosa;
DO I = 1 TO 20;
IF todas{I} = 999 THEN todas{I} = .;
END;
DROP I;
```

Para poner etiquetas a variables, que aparecerán en salida:

LABEL variable = ' etiqueta ' ... ;

Para asignar formatos o informatos a variables:

FORMAT var ... formato. ... ;

Para categorizar por los valores de una o varias variables:

BY var ... ;

-Se usa en combinación con SET o MERGE.

Otras sentencias[18]

III.6 ESPECIFICACIÓN DE VARIABLES EN PASOS DATA O PROC.

Recapitulando, en los pasos DATA o en los pasos PROC se pueden referenciar las variables de las siguientes formas:

-Por sus nombres separados por blancos.

---

[18] Para cambiar atributos de variables: ATTRIB

Para devolver el control de ejecución: RETURN ;

-Si no se encuentra asociada a una etiqueta, bien por una sentencia LINK, bien por una sentencia FILE, envía el control a la primera sentencia del paso DATA.

Para terminar la ejecución del paso DATA: STOP ;

Para asignar informatos a variables:

INFORMAT var ... formato. ... ;

Para retener valores de variables de una ejecución a otra del paso DATA: RETAIN

Para acumular valores de una ejecución a otra del paso DATA: Sentencia de suma:

var + expresion ;

-Se va asociando a la variable var el resultado de la expresión en cada una de las observaciones. var es valor acumulado de varias ejecuciones de paso DATA.

-Con un guión (-) entre dos variables, si existe una secuenciación numérica en el nombre de las variables. (Ej: alt14-alt23).
-Con dos guiones (--) entre dos variables, para indicar todas las variables entre la primera y la segunda. El orden en que se encuentran las variables en un SAS-FILE, viene dado por el orden de aparición de los nombres en el paso DATA.

Existen a esto algunas excepciones:

-En la opción de SAS-FILE, RENAME no se permite referenciar varias variables, antes o después del signo igual. Se debe poner una lista de tríos del tipo: nombre-viejo = nombre-nuevo .
-En la sentencia RENAME del paso DATA, tampoco está permitido.

Ejemplo:

```
DATA ejemplo;
INFILE 'ejemplo.datos';
INPUT  pac  sexo  edad  peso  altura  htamin  htamax  deporte  pulso1
pulso2;
bmi  =  peso**2 / altura;
```

Una vez nombradas las variables como arriba, son equivalentes:
A)
```
pac  sexo  edad  peso  altura  htamin  htamax  deporte  pulso1  pulso2
bmi
```
y
```
pac –- bmi
```

B)
```
edad  peso  altura  htamin  htamax  pulso1  pulso2  bmi
```
y
```
edad – htamax    pulso1 –- bmi
```

IV. MANIPULACIÓN DE CONJUNTOS DE DATOS SAS®.

Creados conjuntos de datos SAS, éstos pueden ser analizados por procedimientos SAS.

Los conjuntos de datos SAS, además, pueden ser leídos en pasos DATA para su transformación en otros conjuntos de datos SAS. Para ello se utilizan las sentencias SET y MERGE.

Entre la operaciones posibles, con SAS-FILEs en los pasos DATA, se encuentran:
Modificar SAS-FILEs.
Concatenar varios SAS-FILEs uno a continuación del otro.
Entremezclar varios SAS-FILEs por los valores de una variable.
Concatenar varios SAS-FILEs uno al lado del otro

IV.1 SENTENCIA SET. UTILIDADES

La sentencia SET:

-Lee observaciones de uno o varios SAS-FILEs.

-No puede combinarse en un mismo paso DATA con sentencias INFILE, MERGE o UPDATE.

-Pueden especificarse varias sentencias SET referidas o no a un mismo SAS-FILE.

-Cada sentencia SET aplicada al mismo SAS-FILE lleva su propio puntero de localización.

-Si se usa en combinación con la sentencia BY, ésta debe aparecer inmediatamente de la sentencia SET. Además los SAS-FILEs, deberán estar ordenados por los valores de las variables BY, que deberán ser las mismas en todos los SAS-FILEs implicados.

Sintaxis:

```
SET  SAS-file  [ ( opcion-SAS-file ... ) ] ...  opcion-set  ;
```

Utilidades de la sentencia SET:

-Realizar una copia de un SAS-FILE para eliminar variables u observaciones, crear nuevas variables o transformar las ya existentes.

-Concatenar varios SAS-FILEs uno debajo de otro. Si el conjunto de variables no coincide, se pondrán a desconocido las variables de cada conjunto que inicialmente no existieran.

```
SET A B;
```

-Entremezclar varios SAS-FILEs, por los valores de alguna variable.

```
SET A B ;
BY var ... ;
```

-Esto es equivalente al punto anterior, sólo que se realiza por los valores de las variables especificadas en BY.

-Concatenar varios SAS-FILEs uno al lado del otro. Si coincide alguna variable, en el SAS-FILE creado, ésta tomará el valor que tomara en el último SAS-FILE especificado.

```
SET A;
SET B;
```

-Si el número de observaciones de los SAS-FILEs es distinto, la ejecución del paso DATA finaliza cuando el número de ejecuciones coincida con el menor número de observaciones en un SAS-FILE.

-El caso anterior, con una sentencia BY. Es equivalente al anterior, sólo que realizado por los valores de las variables del BY.

IV.2 SENTENCIA MERGE. UTILIDADES

La sentencia MERGE:

-Combina observaciones procedentes de dos o más SAS-FILEs en una nueva observación. La forma de combinarlas depende de la presencia o no de una sentencia BY.

-No puede combinarse en un mismo paso DATA con sentencias INFILE, SET o UPDATE.

-Si se usa en combinación con la sentencia BY, ésta debe aparecer inmediatamente de la sentencia MERGE. Además los SAS-FILEs, deberán estar ordenados por los valores de las variables BY, que deberán ser las mismas en todos los SAS-FILEs implicados.

Sintaxis:

```
MERGE  SAS-file  [ ( opcion-SAS-file ... ) ] ...  opcion-merge  ;
```

Utilidades:

-Combinación uno a uno: Se une la primera observación de un SAS-FILE con la primera del otro y así sucesivamente. Si coincide alguna variable, en el SAS-FILE creado, ésta tomará el valor que tomara en el último SAS-FILE especificado.

```
MERGE  A  B ;
```

-Si el número de observaciones de los SAS-FILEs es distinto, la ejecución del paso DATA finaliza cuando el número de ejecuciones coincida con el mayor número de observaciones en un SAS-FILE.

-Si en uno de los conjuntos sólo hay una observación para los valores de las variables var ... , ésta se unirá longitudinalmente con todos las observaciones del otro conjunto con esos valores de variables var...

```
MERGE  A  B ;
BY  var ... ;
```

IV.3 OPCIONES DE LOS CONJUNTOS DE DATOS SAS®

Los conjuntos de datos SAS, tienen unas opciones que le son propias y que se pueden utilizar encerradas entre paréntesis al lado del SAS-FILE, siempre que éstos aparezcan, tanto en pasos DATA como en pasos PROC. Cabe destacar:

```
DATA SAS-file ( opcion-SAS-file ... ) ... ;
SET SAS-file ( opcion-SAS-file ... ) ... ;
MERGE SAS-file ( opcion-SAS-file ... ) ... ;
PROC  nombre-proc DATA = SAS-file ( opcion-SAS-file ... ) ... ;
```

OUTPUT ..... = SAS-file (opcion-SAS-file ...) ;

opcion-SAS-file puede ser:

KEEP = var ...

DROP = var ...

RENAME = (nombre-viejo = nombre-nuevo ...) No permitido listas de variables.

TYPE = tipo indica tipo de conjunto de datos: DATA, CORR, COVA...

usadas únicamente cuando se lee de un SAS-FILE ya existente.

WHERE = condicion

FIRSTOBS = n OBS = n no se pueden usar en combinación con la opción WHERE = o con una sentencia WHERE en el mismo paso DATA o PROC.

IN = var Sólo se puede utilizar en un paso DATA (Sentencias SET, MERGE, UPDATE). Se crea una variable lógica que toma el valor verdadero si se está procesando una observación del correspondiente SAS-FILE. Dicha variable no se añade al SAS-FILE que se crea.

Algunos ejemplos

Ejemplo A) En el fichero de entrada, el valor de la variable tipo indica si un registro de datos contiene información acerca de un profesor o acerca de un alumno.

Se crean dos SAS-FILES, uno con la información de los profesores y el otro con la información de los alumnos.

```
DATA  profesor(KEEP= curso profesor)
            alumno (DROP= tipo curso profesor);

INPUT  tipo  $1.  @ ;

IF TIPO = 'c'  THEN DO;
            INPUT curso profesor $;
            OUTPUT profesor;
            END;

ELSE DO;
      INPUT  nombre  $  edad  nota1-nota8;
      OUTPUT alumno;
      END;
```

V. SENTENCIAS DE USO GENERAL.

V.1 SENTENCIAS USADAS EN PASOS DATA Y PROC

-Algunas de las sentencias utilizadas en los pasos DATA, pueden también utilizarse en los pasos PROC.

-Las sentencias que se refieren a características de las variables u observaciones, si se especifican en un paso DATA, permanecen en los SAS-FILEs creados o modificados de forma permanente. Si se especifican en un paso PROC, hacen referencia a la ejecución del paso PROC únicamente. Entre estas sentencias se encuentran:

-FORMAT, LABEL, WHERE, ...

Algunas sentencias de procedimientos.

Sentencias comunes a algunos procedimientos son:

-BY	Se realiza el procedimiento para los valores de la(s) variable(s) especificadas en BY. Las observaciones del SAS-FILE que se analice deben estar ordenadas por los valores de la(s) variable(s) del BY. La ordenación de un SAS-FILE se puede realizar con el procedimiento SORT. La sentencia BY también se puede utilizar en un paso DATA en combinación con una sentencia SET o MERGE o UPDATE.
-VAR	Selecciona variables para analizar en el procedimiento.
-CLASS	Indica variables de clasificación
-WEIGHT	Especifica una variable de peso.
-FREQ	"

La sintaxis de todas ellas es:

PALABRA especificacion-variable ... ;

V.2 SENTENCIAS DE USO GENERAL

Algunas sentencias son de uso general y no es necesario que se encuentren en un paso DATA o PROC. Pueden encontrarse en cualquier lugar del programa y tienen efecto hasta que aparezca otra del mismo tipo.

Especificación de títulos en la salida de resultados:

TITLE[n] 'título n-ésimo' ;   n <= 10

Especificación de notas a pies de página en la salida de resultados:

FOOTNOTE[n] 'nota a pie de página n-ésima';   n <= 10

Deasignación de referencias a librerías SAS:

LIBNAME ref CLEAR ; (Elimina la identificación de la referencia ref)

Especificación de valores de no observación en los datos, correspondientes a variables numéricas:

MISSING carácter ... ;

carácter puede ser un valor alfabético o _

Especificación de **opciones SAS**[19], en general dependientes del sistema operativo:

OPTIONS opcion ... ;

Algunas opciones son:

NODATE no salida de la fecha en la salida de resultados
LS = n longitud de línea n en la salida de resultados
NOCENTER No centrar los títulos en la salida de resultados
ALTLOG = otro-salida-log Copia de la salida de diagnósticos a otra-salida-log, que puede ser TERM , para terminal
FULLSTIMER Recolección de datos estadísticos de rendimiento del sistema. Existen varias opciones de este tipo

Inclusión de parte de programa procedente de otro fichero:

%INCLUDE[20] 'nombre-de-fichero' ;

Sentencia: RUN ;

Indica el fin de un paso DATA o un paso PROC. Se suele usar en el entorno interactivo o de ventanas.

Sentencia: QUIT ;

Se usa en el entorno interactivo o de ventanas, para indicar que las sentencias de un procedimiento que es interactivo, han finalizado.

También se usa para indicar el fin de un procedimiento gráfico.

Otras sentencias[21]

---

19 Si se desean conocer las opciones posibles se debe ejecutar: `PROC OPTIONS; RUN;`

20 Otra forma de uso de esta sentencia es:
%INCLUDE ref ... ;
ref puede ser:
-el primer nombre de un fichero externo de segundo nombre **SAS**.
-una referencia a la que hay que asignar un fichero externo con la sentencia FILENAME.

21 Identificación de referencias a ficheros externos a SAS:

VI. UTILIZACIÓN DE PROCEDIMIENTOS SAS®.

VI.I. PROCEDIMIENTO FORMAT.

El procedimiento FORMAT:

-Crea formatos para que luego puedan ser asociados a variables, asociación realizada en un paso DATA o en un paso PROC, con la sentencia FORMAT.
-La creación de los formatos se puede realizar de forma temporal a la ejecución de un programa SAS, o de forma permanente, para luego ser utilizados por otros programas (sin necesidad de que se creen los formatos otra vez).
-Para crear un formato SAS de forma permanente basta especificar la opción LIB = LIBRARY[22] en la sentencia PROC FORMAT. En este caso, debe crearse previamente la librería LIBRARY.
-El uso de un formato SAS permanente en otra sesión SAS, requiere que esté accedida la librería LIBRARY.

Sintaxis:

```
PROC  FORMAT  [ LIB = LIBRARY ]22 ;
        VALUE  [$]nombre    especificacion-valor = 'descodificacion' ...  ;  ...
```

FILENAME ref 'nombre-fichero' ;

Deasignación de referencias a ficheros:

FILENAME ref CLEAR ;(Elimina la identificación de la referencia ref)
Ejecución de comandos del sistema operativo:

X ['] comando-sistema-operativo ['] ;

Ir al entorno del sistema operativo:

X;

Para volver a la continuación de ejecución del programa SAS basta con pulsar ^D.

[22] Se pueden almacenar formatos permanentes en librerías distintas de la librería LIBRARY, por ejemplo Nombre_librería. Para ello es necesario
Tener accedida la librería Nombre_librería
La sentencia PROC FORMAT LIB = nombre_librería ;
Cuando un SAS-FILE permanente tiene asociados formatos en un librería Nombre_librería, antes de acceder al SAS-FILE es necesario:
Tener accedida la librería Nombre_librería
Permitir a SAS que busque formatos en la librería Nombre_librería. Esto se hace con la sentencia FMTSEARCH = (WORK LIBRARY Nombre_librería) ; o bien con la sentencia FMTSEARCH = (WORK LIBRARY Nombre_librería). La sentencia FMTSEARCH marca un orden de búsqueda de formatos en las librerías (por defecto en SAS este orden es WORK LIBRARY)

-Si el formato se asociará a una variable carácter, deben especificarse el dolar ($) justo antes del nombre del formato y las comillas para encerrar especificaciones de valor.

-especificacion-valor puede ser:

val — indica el valor val.

vali-valf — indica desde valor vali hasta valor varf, ambos incluidos. vali puede ser LOW para indicar el mínimo de los valores posibles y vali puede ser HIGH, para indicar el máximo.

vali<-valf — ... excluyendo vali. Valf puede ser HIGH.

vali-<valf — ... o excluyendo valf. Vali puede ser LOW.

varias combinaciones anteriores, separadas por comas (,).

en el caso de formatos para variables carácter, se han de especificar los valores encerrados en comillas (').

other — indica cualquier otro valor no especificado en las especificaciones anteriores.

Ejemplo:

```
PROC FORMAT;
      VALUE  sino  1 = 'si'  2 = 'no';
      VALUE  $sexo  'm' = 'mujer'  'v' = 'varon';
      VALUE  nota  LOW-5 = 'suspenso'  5<-7 = 'aprobado'
            7<-HIGH = 'excelente';

DATA alumno;
INFILE 'D:\datos';
INPUT id genero $ nota p1-p7;
FORMAT genero $sexo. nota nota. p1-p7 sino.;
RUN;
```

VI.II. OTROS PROCEDIMIENTOS

PROCEDIMIENTO PRINT.

PROC PRINT [DATA = SAS-file] ;
[VAR especifivacion-variable ... ;]
[BY especificacion-variable ... ;]
[SUM especificacion-variable ... ;]

PROCEDIMIENTO SORT.

PROC SORT [DATA = SAS-file] ;
BY [DESCENDING] especificacion-variable ... ;

En general, orden ascendente de SAS: _, ., a-z, -n, 0, +n   siendo n un número.

PROCEDIMIENTO CONTENTS.

PROC CONTENTS [DATA = SAS-file] [POSITION] ;

Muestra información sobre el SAS-FILE. La opción POSITION hace que las variables aparezcan en el orden del SAS-FILE y no en orden alfabético.

PROCEDIMIENTO SQL.

Procedimiento que utiliza el lenguaje SQL para la creación o manipulación de tablas, vistas, etc...

PROCEDIMIENTO MEANS

PROC MEANS [DATA = SAS-file] opcion... ;
[VAR especifivacion-variable ... ;]
[CLASS especificacion-variable ... ;]
[WEIGHT especifivacion-variable ... ;]
[FREQ especificacion-variable ... ;]
[BY especificacion-variable ... ;]
[OUTPUT OUT = SAS-file opcion ... ;]

Ejemplos del proc means puende ser:

<u>Ejemplo</u>

```
PROC MEANS DATA=ejercicio N MEAN STD STDERR P10 P25 MEDIAN MIN MAX;
Var edad pulso1 pulso2 pulsom;

PROC MEANS DATA=sasuser.class
     CLASS sex age;
      TYPE sex*age;

PROC MEANS DATA=airline.fltaten;
      CLASS state jobClass genderCode;
      WAYS 1 2;
RUN;
```

La sentencia TYPE permite especificar las combinaciones de las variables de clase.
La sentencia WAYS: especifica uno o más enteros que define el número de variables de clasificación a combinar para formar todas las combinaciones posibles de esas variables de clasificación.
Permite especificar múltiples sentencias CLASS.

PROCEDIMIENTO TTEST

PROC TTEST [DATA = SAS-file] opcion... ;
[VAR especifivacion-variables ... ;]
[CLASS especificacion-variable ... ;]
[PAIRED especificacion-variables ... ;]
[WEIGHT especifivacion-variable ... ;]
[FREQ especificacion-variable ... ;]
[BY especificacion-variables ... ;]

PROC TTEST < *options* > **;**

The following options can appear in the PROC TTEST statement.

- ALPHA=*p*   /* Intervalos confidenciales 100(1-*p*)% ; Por defecto p = 0.05 */
- CI=EQUAL
- COCHRAN
- DATA=*SAS-data-set*
- H0=*m*

CI=EQUAL
CI=UMPU
CI=NONE

The CI= option specifies whether a confidence interval is displayed for and, if so, what kind. The CI=EQUAL option specifies an equal tailed confidence interval, and it is the default. The CI=UMPU option specifies an interval based on the uniformly most powerful unbiased test of . The CI=NONE option requests that no confidence interval be displayed for . The values EQUAL and UMPU together request that both types of confidence intervals be displayed. If the value NONE is specified with one or both of the values EQUAL and UMPU, NONE takes precedence

Ejemplo:

```
PROC TTEST;
VAR   v1 ;
CLASS sexo;
```

```
                              The TTEST Procedure

                                   Statistics

                        Lower CL          Upper CL  Lower CL                   Upper CL
Variable  Class       N     Mean     Mean     Mean   Std Dev  Std Dev  Std Dev  Std Err

V1        hombre    138   182.06   183.85   185.65    9.5293   10.655   12.086   0.907
V1        mujer      92   173.11   175.58   178.06     10.44   11.953   13.983  1.2462
V1        Diff (1-2)      5.3044   8.2725   11.241    10.252   11.191   12.322  1.5063


                                    T-Tests

             Variable    Method           Variances      DF    t Value    Pr > |t|

             V1          Pooled           Equal         228       5.49      <.0001
             V1          Satterthwaite    Unequal       179       5.37      <.0001


                            Equality of Variances

               Variable    Method      Num DF    Den DF    F Value    Pr > F
               V1          Folded F        91       137       1.26    0.2224
```

```
PROC TTEST CI= NONE;
VAR   v1 ;
CLASS sexo;
```

No muestra en salida los intervalos de confianza.

```
DATA varon;
SET duda;
IF sexo='hombre';
PROC TTEST;
VAR   v1 ;
RUN;
```

```
                    Lower CL              Upper CL   Lower CL             Upper CL
 Variable       N    Mean       Mean        Mean     Std Dev    Std Dev    Std Dev    Std Err

 V1           138   182.06    183.85      185.65      9.5293     10.655     12.086      0.907

                                              T-Tests

                              Variable       DF     t Value     Pr > |t|

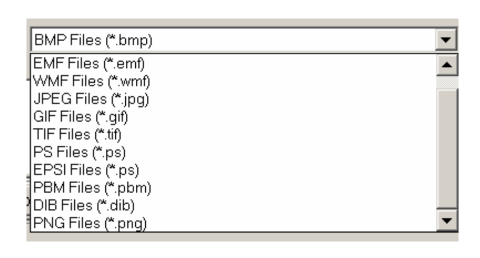
                              V1            137      202.69       <.0001
```

PROCEDIMIENTOS FREQ, PLOT, CHART, UNIVARIATE, TRANSPOSE

VI.III. GRAFICOS EN SAS

Los gráficos producidos en SAS, por defecto se muestran en el dispositivo de salida de SAS en la ventana Graph.

Se pueden almacenar gráficos en ficheros con otros formatos. Esto se puede realizar desde la ventana Graph mediante la opción de menú *Archivo/Exportar como Imagen* y seleccionar alguno de los formatos que aparecen en la figura.



o bien mediante código SAS con la sentencia goptions especificando el dispositivo gráfico y el fichero antes del procedimiento que produce el gráfico.

```
FILENAME referencia-fichero 'unidad-directorio-fichero';

GOPTIONS   DEVICE=dispositivo   GFSNAME=referencia-fichero   GFSMODE=replace
[otras-opciones];
```

Dispositivos pueden ser: `psepsf` para encapsulado postcript, `cljpsa4` para postcript, gif para formato gif, , etc, etc...

Otras-opciones incluye:

```
ROTATE=LANDSCAPE
```

La calidad de los gráficos de los dispositivos de SAS no es óptima salvo para el caso del propio dispositivo de SAS y los formatos postcript y encapsulado postcript.

Ejemplo:

```
FILENAME file1 'c:\graficos\ejemplo.eps';

GOPTIONS DEVICE=psepsf GFSNAME=file1 GFSMODE=replace ;

PROC GPLOT DATA=ejer;
PLOT pulso2*pulso1=sexo;
```

En el ejemplo, el gráfico se muestra en la ventana graph y se almacena con formato encapsulado postcript en el fichero c:\graficos\ejemplo.eps.

Antes de obtener un nuevo gráfico se deben especificar de nuevo las sentencias filename y goptions correspondientes. Si se desea eliminar la información de un dispositivo específico, se debe especificar la sentencia:

```
GOPTIONS RESET=ALL;
```

Es importante, para evitar problemas de que SAS se quede enganchado con un procedimiento gráfico, que después de cada procedimiento que realiza un gráfico se añadan las dos sentencias:

```
RUN;
QUIT;
```

Algunos procedimientos estadísticos pueden producir gráficos. Para ello hay que consultar la sintaxis de los procedimientos y/o la documentación de los mismos.

Ejemplo

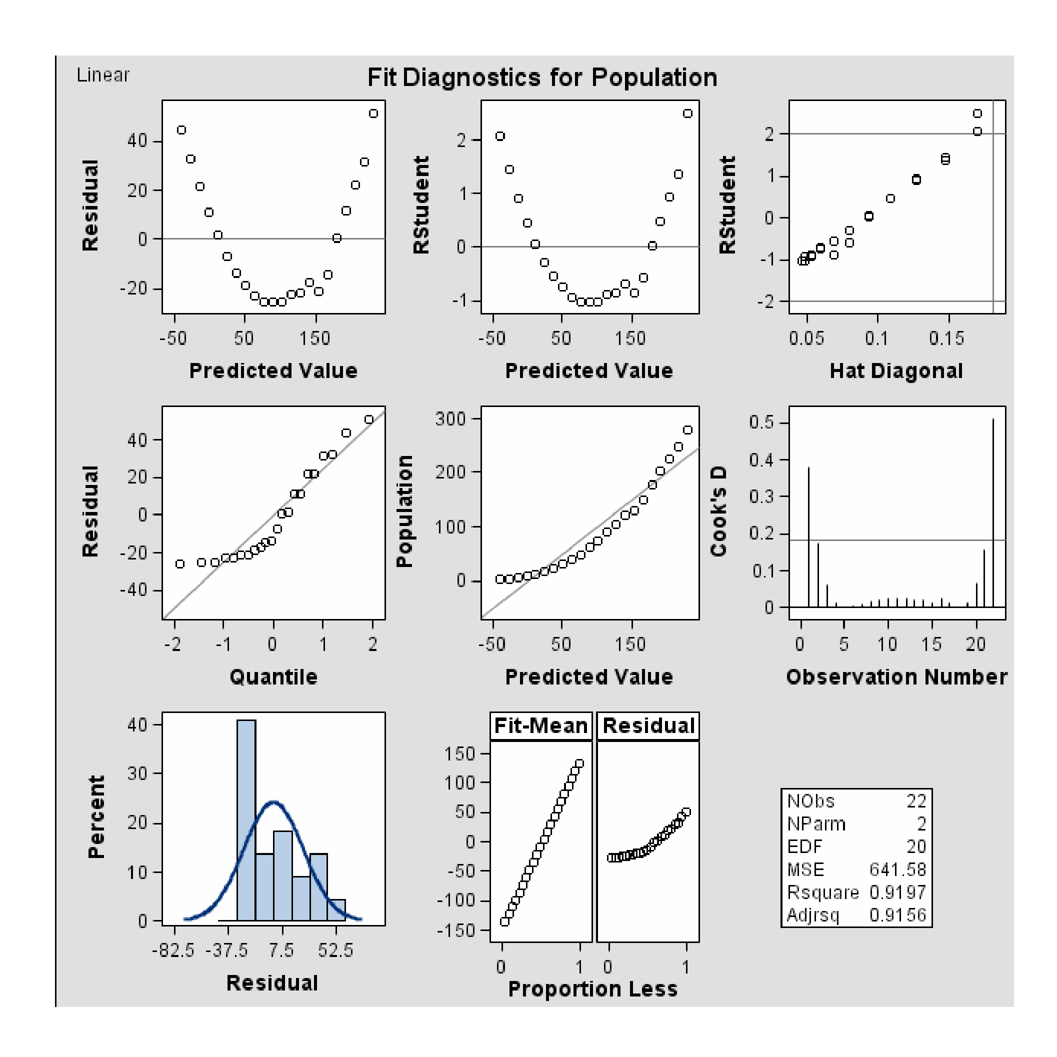
```
PROC REG DATA= ejer ;
MODEL pulso2=pulso1;
PLOT pulso2*pulso1 / PRED ; * PLOT pulso2*pulso1;
PLOT r.*p.;
RUN; QUIT;
```

En SAS v9 se encuentra en período de pruebas la **salida de gráficos con ODS,** incorporada a muchos procedimientos estadísticos. Esta salida se puede realizar en varios formatos.

Ejemplo (salida ODS en SAS v9)

```
ODS HTML;
   ODS GRAPHICS ON;
    PROC REG DATA=uspopulation;
      LINEAR: MODEL population=year;
   RUN; QUIT;
    ODS GRAPHICS OFF;
   ODS HTML CLOSE;
```

A continuación algunos gráficos producidos por las últimas sentencias.



SAS/GRAPH consta de procedimientos gráficos.

Ejemplo

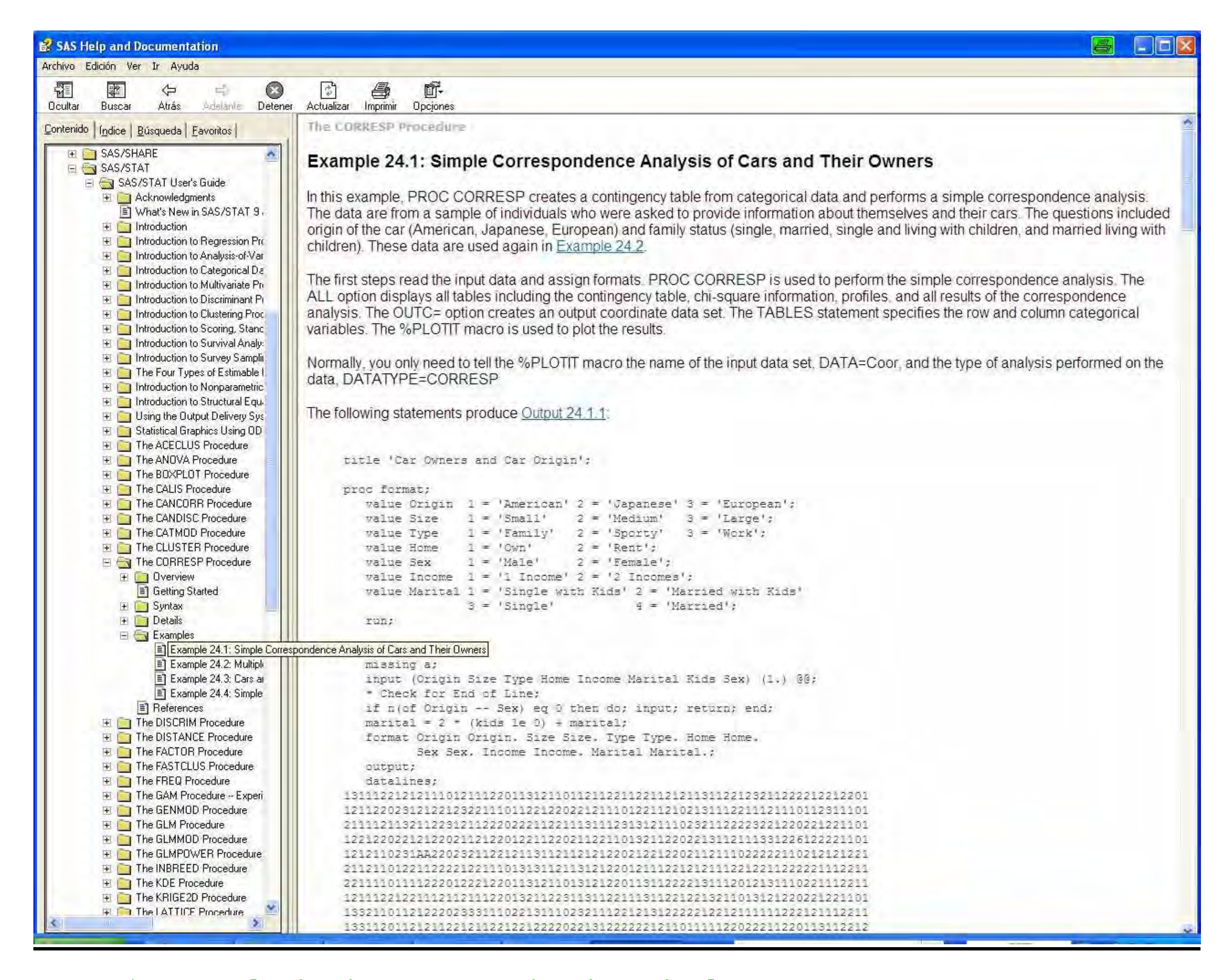
```
PROC GPLOT DATA=ejer;
PLOT pulso2*pulso1=sexo;
RUN; QUIT;

PROC GCHART DATA=ejer;
HBAR pulso1;
VBAR pulso1 /GROUP=sexo;
RUN; QUIT;
```

También existen macros que vienen con el producto que se pueden utilizar para producir gráficos, por ejemplo, la macro plotit realiza gráficos de puntos en un plano X, Y. Permite etiquetar los puntos, dar diferentes tamaños y colores dependiendo de una tercera variable, etc… Un uso particular de la misma es la representación de puntos y variables sobre los ejes factoriales en las técnicas multivariantes factoriales.

Ejemplo

(Ejemplo 24.1, que se encuentra en la ayuda de SAS/STAT → SAS/STAT User's Guide → The CORRESP procedure)




```
*--- Analysis de Correspondencias Simples ---;
PROC CORRESP ALL DATA=cars OUTC=COOR;
   TABLES marital, origin;

%PLOTIT(DATA=COOR, DATATYPE=CORRESP)

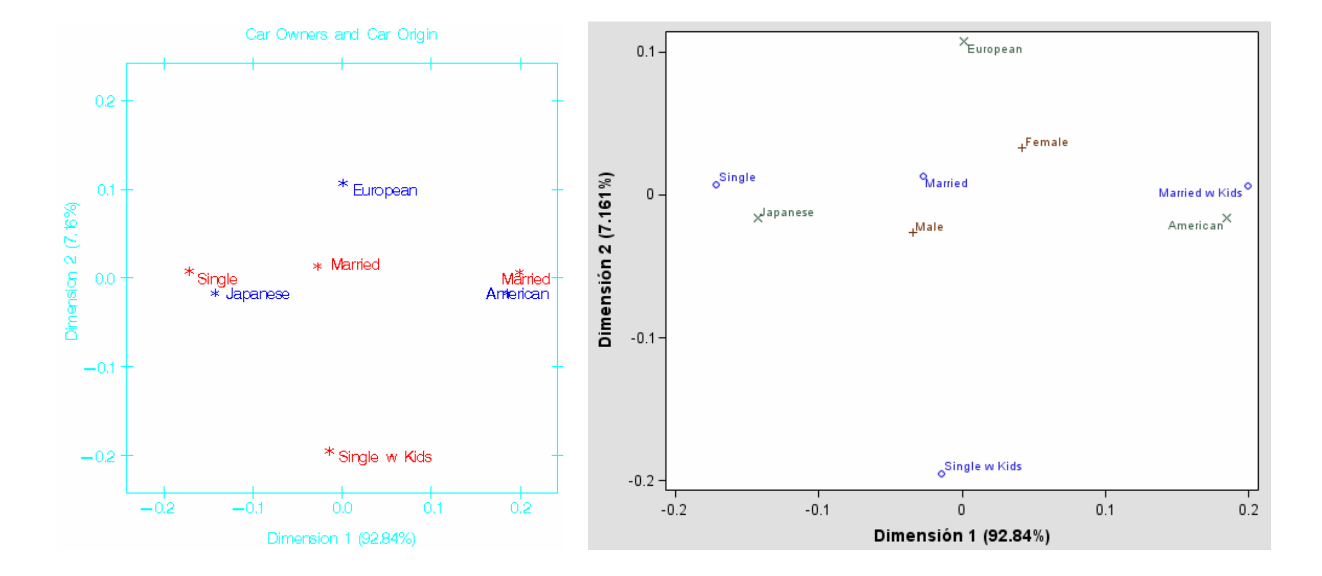
/* desde la version 9 se encuentra en período de pruebas la salida de
gráficos asociadas a procedimientos, mediante ODS */
```

En la ayuda de SAS se pueden encontrar más ejemplos de uso de la macro plotit. Además buscando plotit en los directorios de sas se encuentran los ficheros plotit.sas (la macro) y plotitex.sas (ejemplos de uso de la macro).

Ejemplo(salida ODS en SAS v9):

```
ODS HTML;
   ODS GRAPHICS ON;
   *---PERFORM SIMPLE CORRESPONDENCE ANALYSIS---;
   PROC CORRESP SHORT DATA=cars;
      TABLES sex marital, origin;
      SUPVAR sex;
   RUN;
   ODS GRAPHICS OFF;
   ODS HTML CLOSE;
```

A continuación se muestra la salida de gráficos con la macro %plotit y con la salida mediante ODS.



Ejemplo

```
%PLOTIT(DATA=fact2, DATATYPE=, PLOTVARS=factor2 factor1,
LABELVAR=ident, SYMVAR=., SYMSIZE=0.4,PAINT=crel_eje1_punto )
```

En este ejemplo, el SAS-file `FACT2` no es un sasfile producido por un procedimiento anterior.

VII. UTILIDADES DE LA PROGRAMACIÓN SAS® E INTERACCIÓN CON OTROS PAQUETES.

VII.1 (AVANZADO) PROGRAMACIÓN CON MACROS

Una *macro* es un conjunto de sentencias *macro* y/o sentencias SAS, que puede ser referenciada por otras partes del programa. Se permite, asimismo, la especificación de argumentos. Dichos argumentos se llaman *macro-variables*.

Veamos un escueto resumen:

Creación de *macros*.

Se realiza mediante:

```
%MACRO  nombre-macro  [( par1 [, par2 ... ] )]  ;
(Sentencias  macro  y/o  sentencias SAS)
%MEND  [ nombre-macro ]  ;
```

-Los parámetros de la *macro* son *macro variables locales*, que tienen validez únicamente en esa *macro*.

-La referencia dentro de la *macro* a parámetros de la misma, requiere un ampersand (&) justo delante de los mismos.

Macro variables.

Las *macro variables* son variables que transfieren su valor a lo largo de varios pasos DATA o pasos PROC.

Existen *macro variables globales* y *macro variables locales*. Las *macro variables globales* tienen valor en un programa desde su creación. Las *macro variables locales* tienen valor unicamente durante la ejecución de una *macro*.

Creación de *macro variables*

Se pueden crear en cualquier lugar de un programa SAS con la sentencia:

%LET macro-variable [= valor-inicial] ;

Si una *macro variable* se crea dentro de una *macro*, es una *macro variable local* a esa *macro*.

Si el valor que se quiere asignar a una *macro variable* es un conjunto de sentencias SAS, esto se realiza mediante el uso de %STR:

%LET macro-variable = %STR (sentencias SAS);

Referencia a *macro variables*.

La referencia de *macro variables* se hace siempre con un & delante de su nombre.

Llamada de *macros*.

%nombre-macro [(val1 [, val2...])] ;

La llamada a una *macro* debe encontrarse después de su creación.

Se pueden hacer llamadas a *macros* dentro de la definición de una *macro*.

Ejemplo.

```
%MACRO  unavar (datos, x) ;
PROC  GLM  DATA = &datos  ;
CLASS  sexo ;
MODEL  &x  =  sexo  ;
%MEND  unavar;

%unavar  (ejercici, pulso) ;
```

Sentencias *macro* en la definición de una macro.

En la definición de una macro se pueden especificar las siguientes sentencias macro:

%DO *macro-variable* = num1 %TO num2 [%BY num3] ;
(Sentencias macro y/o sentencias SAS)
%END ;

En la sentencia %DO, la *macro-variable*, que se crea, no lleva &. Dentro del bucle, para referirse a la *macro-variable*, debe especificarse *&macro-variable*.

%DO [%WHILE | %UNTIL] condición ;
(Sentencias macro y/o sentencias SAS)
%END ;

%IF condición %THEN accion; [%ELSE accion2 ;]

%GOTO etiqueta ;

%etiqueta : [sentencia] ;

%STR (parte de sentencias SAS y/o sentencias SAS) ;

Para especificar una cadena de caracteres.

%INPUT macro-variable ;

Para entrada de datos del exterior.

Sentencias *macro* en la definición de una *macro* o externas a la definición.

%LET macro-variable [= valor-inicial] ;

Crea una *macro variable* a la que da un determinado valor inicial. Si se especifica la sentencia fuera de cualquier *macro*, tiene validez para todo el programa SAS. En caso, contrario, para la *macro* en que se ha definido.

%PUT texto;

Permite la escritura de un texto.

Ejemplo.

```
%LET  num = 30;
%MACRO  TOTAL ;
%DO  i = 1  %TO  &num ;
      %INPUT  cen ;
      PROC  TABULATE  DATA = &cen ;
      (Sentencias SAS que terminen de especificar un programa)
%END ;
DROP i;
%MEND ;

%TOTAL ;  /* se realiza la ejecución de la macro */
```

Combinación de *macro variables*:

Combinación de los valores de dos *macro variables*:

&macro-variable1&macro-variable2

Combinación de las valores de dos *macro variables* con intercalación de otro nombre, digamos X:

&macro-variable1.X&macro-variable2

Creación de *macro variables* con valores de un paso data:

Call SYMPUT ('macro-variable', variable);

Se crea la *macro-variable*, si no existía, y se le da el valor de *variable*
Esta sentencia no es necesario referenciarla en la definición de una *macro.*

Asignación de valores de una *macro variable* a variables de un paso DATA:

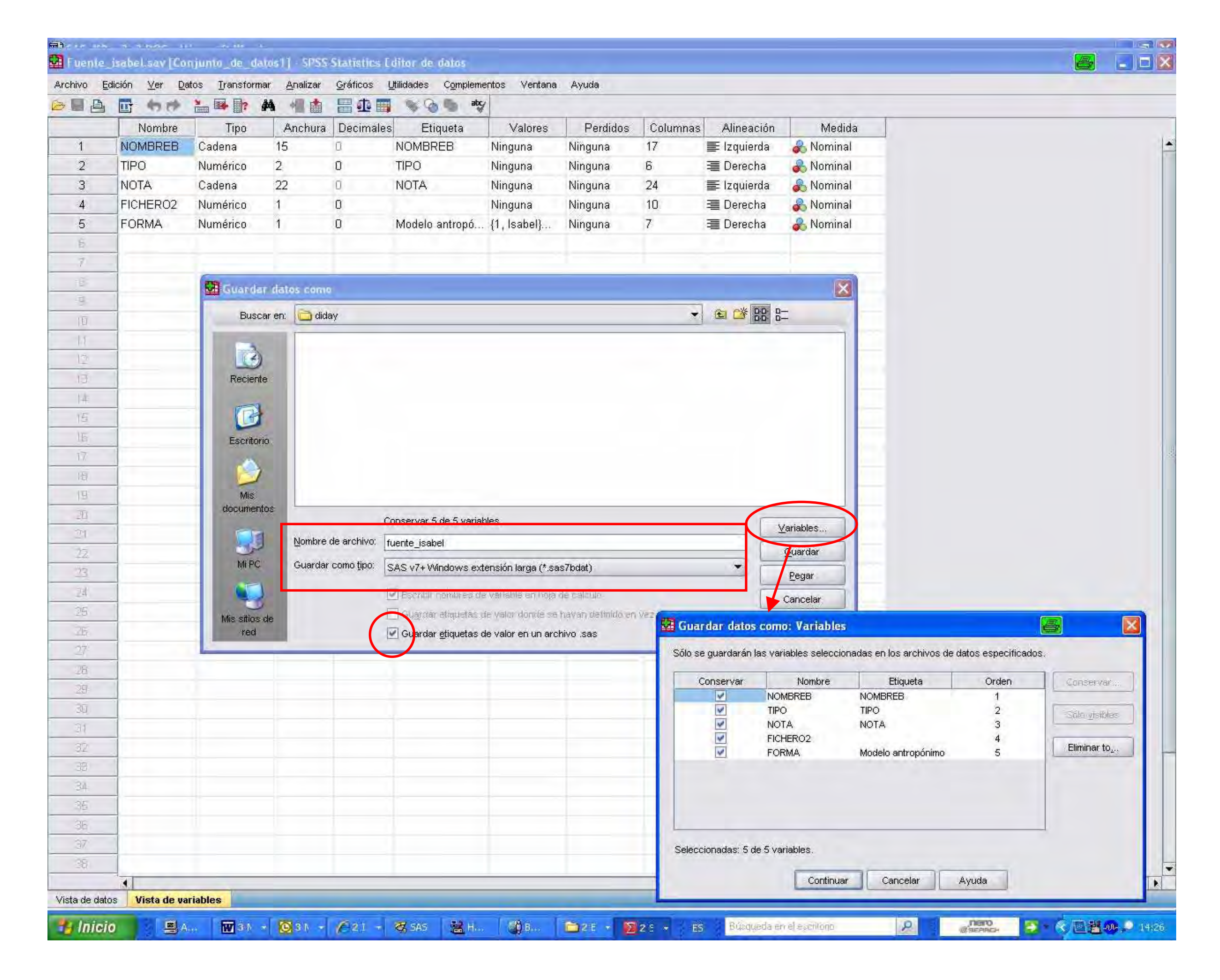
Se realiza mediante la llamada a la función:

SYMGET('macro-variable');

Esta sentencia no es necesario referenciarla en la definición de una *macro.*

VII.2 INTERACCIÓN ENTRE ENTRE SPSS Y SAS®

Existen varias formas de pasar de un SPSS-FILE a un SAS-FILE. La que creemos es más cómoda es hacer la conversión en SPSS. Activado el SPSS-FILE, seleccionar la opción de menú *Archivo→Guardar como* y en la ventana que aparece se debe elegir que sea del tipo sas7bdat. Al activar la pestaña Variables se pueden deseleccionar las que no interese guardar. Si en SPSS se han añadido etiquetas a las variables o a las categorías, debe seleccionarse la opción *Guardar etiquetas de valor en un archivo .sas*.



Esta operación genera los ficheros nombre_fichero.sas7bdat y opcionalmente nombre_fichero.sas. en el ejemplo el contenido del fichero Fuente_isabel.sas es el siguiente.

```
LIBNAME LIBRARY 'd:\diday\' ;

PROC FORMAT LIBRARY = library ;
   VALUE
      1 = 'Isabel'
      2 = 'Hipocorístico Isabel'
      3 = 'C. interna con Isabel y derivados'
      4 = 'Elizabeth y otros'
      5 = 'Hipocorístico extranjero'
      6 = 'C. Interna con Elizabeth y otros' ;

PROC DATASETS LIBRARY = library ;
MODIFY fuente_isabel;
   FORMAT     forma.;    ***  atención spss 17 deja un blanco entre forma y .
                              que debe eliminarse;
QUIT;
```

que asocia la librería LIBRARY con el directorio d:\diday y asocia los formatos al fichero de datos SAS library.fuente_isabel. Otras formas de conversión en el pie de página[23].

[23] **Conversión de un SPSS-FILE a un SAS-FILE. Procedimiento CONVERT.**

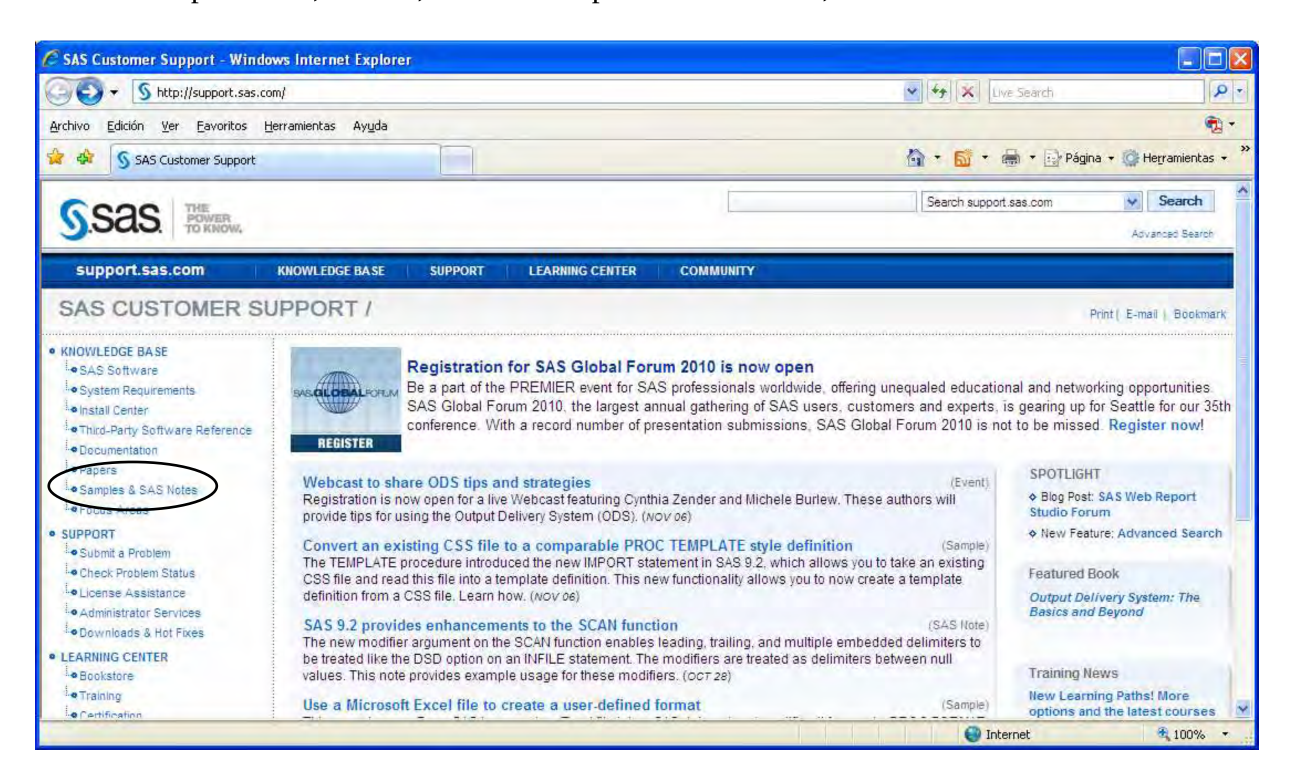
VII.3 EXPORTACIÓN/IMPORTACIÓN DE FICHEROS SAS® A OTRAS PLATAFORMAS

Esto se realiza con los procedimientos **CPORT** y **CIMPORT**.

VII.4 MACROS DE SAS EN URL DE SAS Y CON EL SOFTWARE SAS

Base de datos en la url de SAS

En la dirección url *http://support.sas.com/* se puede acceder a una base de datos con información detallada de productos, macros, soluciones a problemas de SAS, etc…

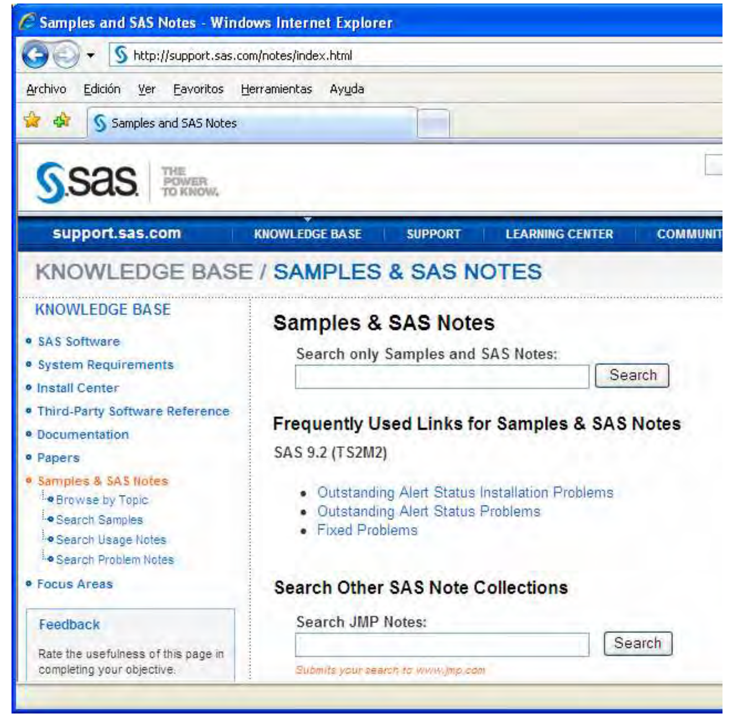


Cabe destacar la pestaña de la izquierda “Samples and SAS notes”:

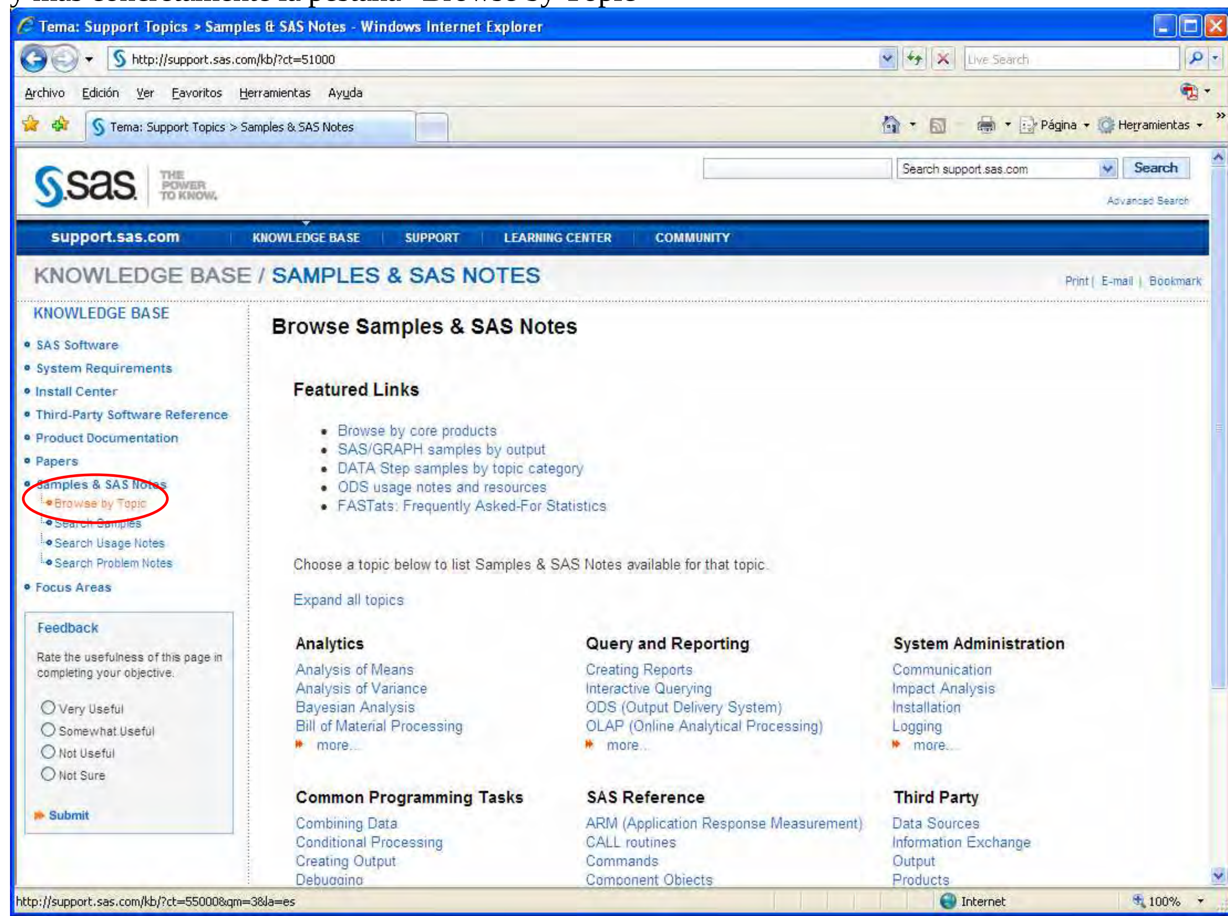
- Existe la posibilidad de analizar conjuntos de datos BMDP (BMDP-FILEs) o SPSS (SPSS-FILEs) en SAS, previa conversión de los mismos a SAS-FILEs.
- Esta conversión se realiza mediante el procedimiento CONVERT.
- Previamente se debe asociar una referencia con el nombre del fichero que contiene el BMDP-FILE o el SPSS-FILE con la sentencia FILENAME.
- Para un fichero spss (que debe exportarse en formato portable (.por) en SPSS), se deben incluir las siguientes sentencias:
  FILENAME referencia 'fichero-que-contiene-el-fichero SPPS’;
  PROC CONVERT SPSS = referencia [OUT = fich2];

Ejemplo:

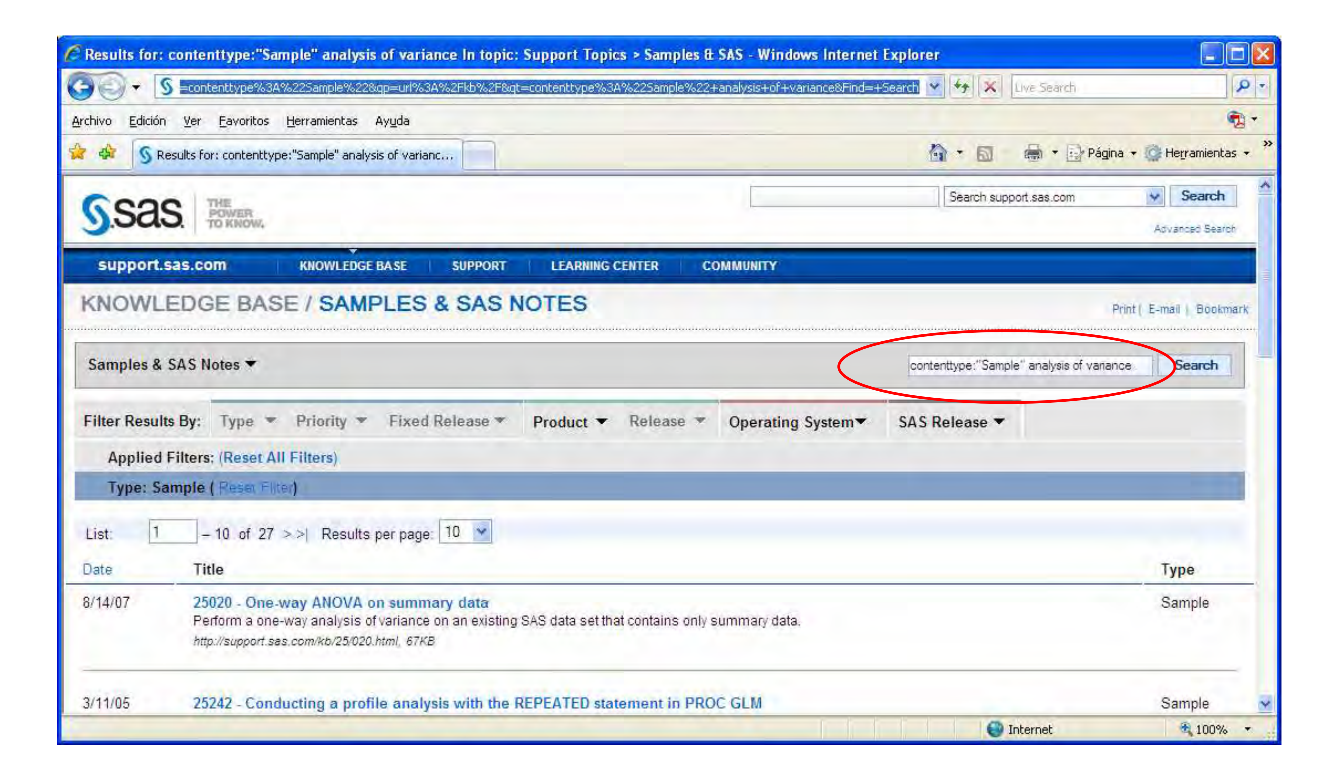
```
FILENAME  base  'd:\usuario\ines\base.por';
      PROC  CONVERT  SPSS = base   OUT = basef ;
```



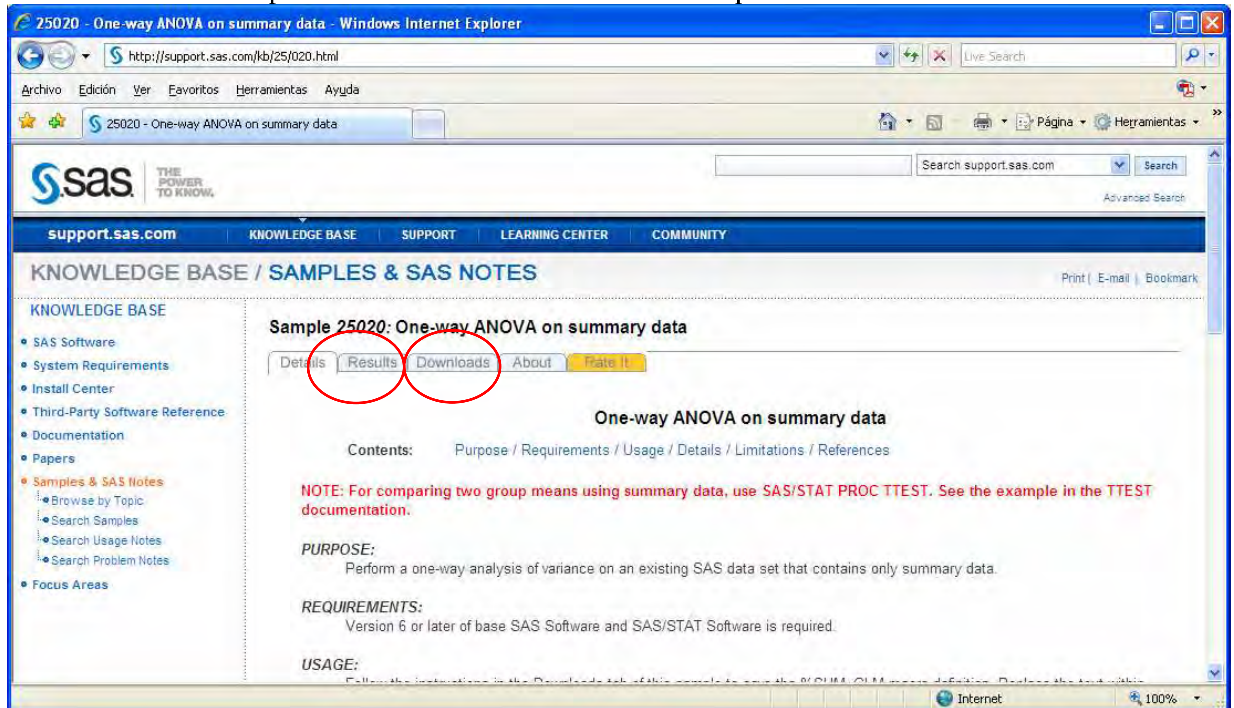
y más concretamente la pestaña "Browse by Topic"



y la pestaña "Search Samples" desde la que se pueden buscar ejemplos de programas SAS que proporcionan macros SAS para poder ser utilizadas por los usuarios del programa. A continuación se buscan ejemplos relativos a "analysis of variance"



Accediendo a uno de los ejemplos, se puede descargar el programa de ejemplo que contiene la macro de SAS correspondiente. Esto se hace a través de la pestaña “Downloads”.



Contenido del fichero sum_glm.sas que se descarga:

```
%macro sum_glm(version,

     /*********************** REQUIRED parameters *************************/
     N=            , /* Name of the variable containing the frequency counts */
                     /* of the individual groups.                            */
     Mean=         , /* Name of the variable containing the means of the     */
                     /* individual groups.                                   */
     StdDev=       , /* Name of variable containing the standard deviations  */
                     /* of the individual groups.                            */
```

```
     Group=       , /* Name of the classification (grouping) variable.      */

     /*********************** OPTIONAL parameter **************************/
     Data=_last_  , /* Name of the SAS data set containing the summary      */
                    /* statistics.  If this parameter is not specified,     */
                    /* the last created dataset will be used.               */
     LSopts =     , /* Any option(s) that are desired on the LSMEANS        */
                    /* statement.                                           */
     By =           /* By variable(s).                                      */
                    );

%if &version ne %then %put SUM_GLM macro Version 1.1;

%let opts = _last_=%sysfunc(getoption(_last_));
%if &data=_last_ %then %let data=&syslast;

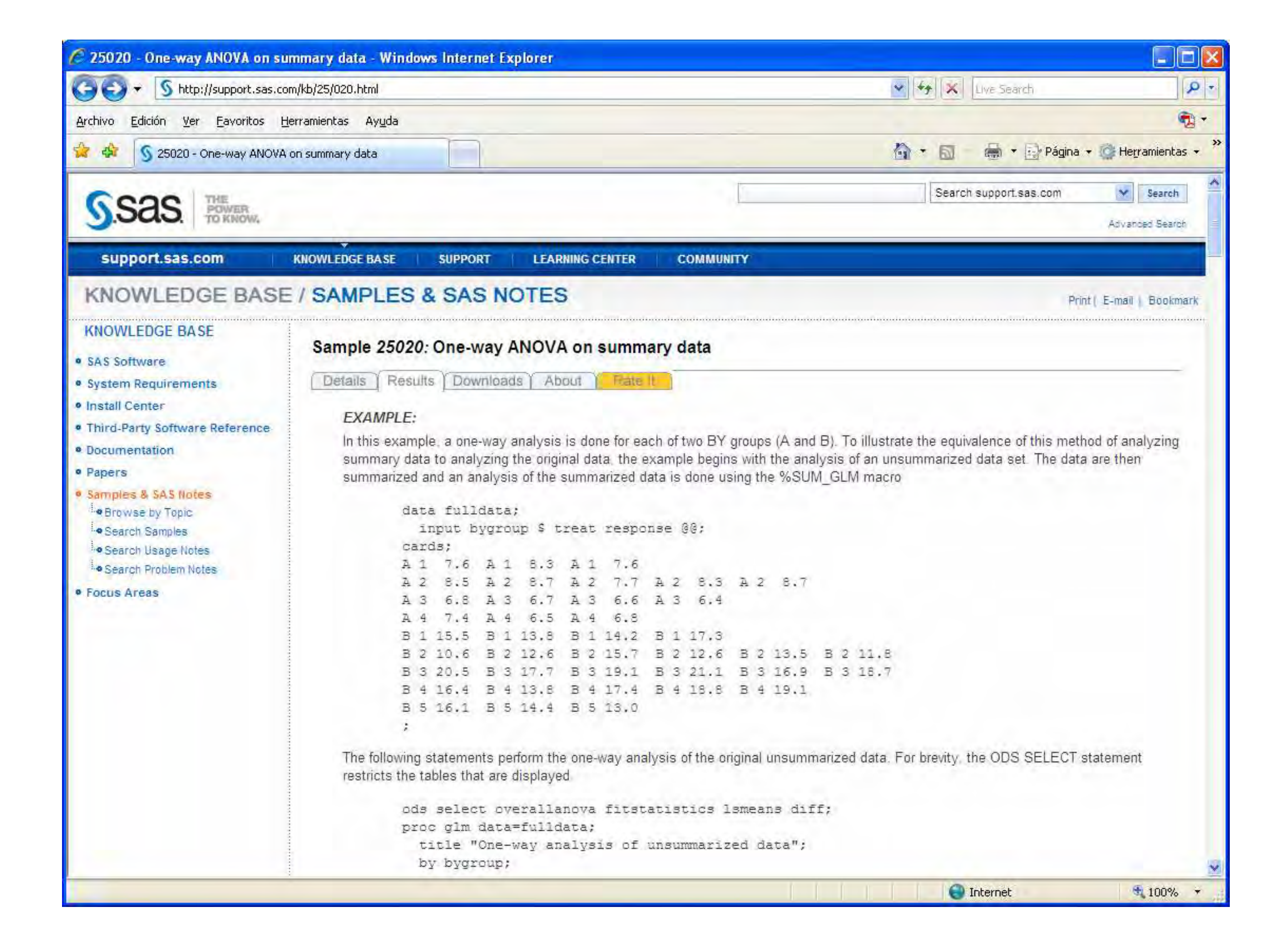
title 'SUM_GLM Macro: Analysis of Variance on Summary Statistics';

/* Create data based on summary statistics */
data _working;
  set &data;
  Xis = &Mean + sqrt((&StdDev**2)/&N);
  Xns = &N*&Mean - (&N-1)*Xis;
  y=Xis;  Freq=&N-1;  output;
  y=Xns;  Freq=1;     output;
  run;

proc glm data=_working;
  by &By;
  class &Group;
  freq Freq;
  model y = &Group;
  lsmeans &Group / &LSopts;
  run;
  quit;

options &opts;
title;
%mend;
```

En la pestaña “Results” se encuentra un ejemplo completo que utiliza la macro SUM_GLM.



Macros en el software SAS

También vienen incorporadas macros[24] con el programa SAS, así por ejemplo en cualquier subcarpeta sample que se incorpore en la instalación de SAS, por ejemplo C:\Archivos de programa\SAS\SAS 9.1\stat\sample.

Existe un documento en versión borrador que detalla la relación de SAS con el análisis estadístico de datos. Se puede solicitar a través del servicio de atención al usuario de los Servicios Informáticos (91 3 94 47 74, solicitarlo a Apoyo a Investigación, preguntando por Carmen Bravo)

[24] Existía antiguamente la macro **TREEDISC** para aplicación de la técnica de árboles de segmentación. Esta macro ya no es soportada por SAS y además pueden obtenerse árboles de segmentación con la solución de SAS, Enterprise Miner.

APÉNDICE I (Platón. Unix)

Ejecución de un programa SAS®.

Ejecución en modo no interactivo (opción recomendada):

La ejecución de forma no interactiva requiere la inclusión del programa SAS en un fichero. Si el segundo nombre es **SAS**, la ejecución se realiza mediante: **SAS** primer-nombre. De dicha ejecución se obtienen los ficheros *primer-nombre.lst* y *primer-nombre.log*, que incluyen respectivamente la salida de resultados si existe y la salida de diagnósticos.

Si el segundo nombre del fichero es distinto de *SAS*, la ejecución se realiza mediante: **SAS** nombre-fichero. De dicha ejecución se obtienen los ficheros *nombre-fichero.lst* y *nombre-fichero.log*, que incluyen respectivamente la salida de resultados si existe y la salida de diagnósticos.

Ejecución interactiva o en modo ventanas:

Sesión X: SAS

en el caso de que no sea una terminal X, debe habilitarse la sesión para habilitarla. Esto se hace con el comando Unix: setenv DISPLAY …….

Modo texto: SAS -fsdevice ascii

Si la ejecución es de forma interactiva, se incluirá el programa en la ventana *editor avanzado*. La salida de resultados, si existe, se incluye en la ventana OUTPUT y la salida de diagnósticos en la ventana LOG.

Directorio de ejemplos:

/usr/local/SAS_8.2/samples

APÉNDICE II. (AVANZADO) SAS-FILES Y SAS-ENTERPRISE GUIDE

Nota Importante

Si no se desean perder los accesos a los perfiles de usuario de SAS y SAS-Enterprise Guide, no deben tenerse abiertas dos sesiones simultáneamente de estos dos programas. El perfil de usuario contiene información acerca de los accesos a librerías permanentes.

SAS-Files creados en SAS®

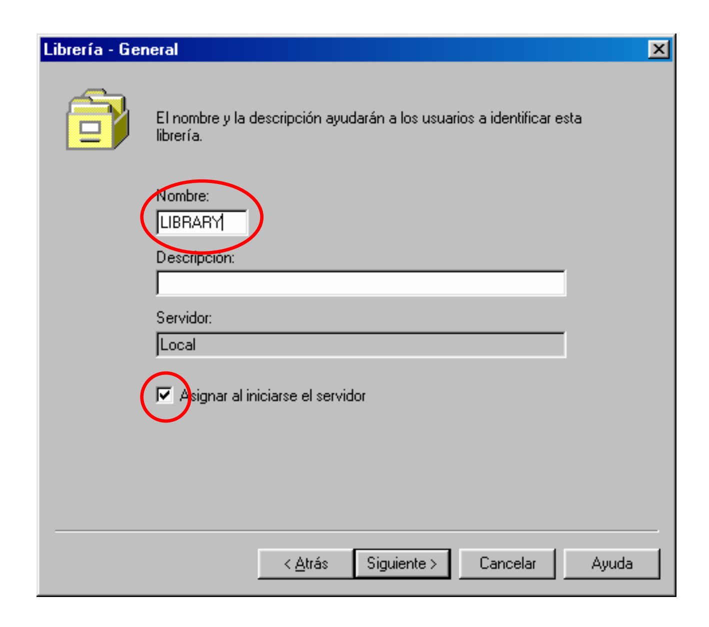
Los SAS-files permanentes creados con SAS pueden leerse en SAS-Enterprise Guide. Si los SAS-Files no tienen asociados formatos (permanentes), el acceso es directo. Si las variables tienen formatos permanentes éstos se encuentran en la librería LIBRARY[25] (véase creación de formatos en esta guía). Esta librería no está por defecto accesible en SAS-Enterprise Guide. Se pueden hacer dos cosas alternativas:

- Copiar los formatos en Windows desde el directorio asociado a la librería LIBRARY hasta el directorio asociado a la librería SASUSER (por defecto[26], es el directorio \Mis documentos\My SAS Files\9.1 en la unidad donde se encuentra instalado SAS).
- Asociar la librería LIBRARY al directorio correspondiente con el programa administrador asociado a Enterprise Guide. Esto se realiza de la siguiente forma:

Asignación de librerías en SAS-Enterprise Guide, desde el programa Administrador

Para crear una librería en SAS-Enterprise Guide, se puede hacer desde el programa Administrador asociado al Eguide, con la opción de menú Archivo→ Nuevo → Librería

[25] Según se explica en este manual. Se pueden estructurar los formatos en otras librerías.
[26] Se puede averiguar el directorio asociado con la librería SASUSER, pulsando sobre ella el botón derecho del ratón y seleccionando propiedades.



Si se selecciona la opción Asignar al iniciarse el servidor, esta librería quedará accesible siempre que se ejecute SAS-Enterprise Guide.

SAS-Files creados en SAS® -Enterprise Guide

Los SAS-files permanentes creados con SAS-Enterprise Guide pueden leerse en SAS. Si los SAS-Files no tienen asociados formatos (permanentes), el acceso es directo. Si las variables tienen formatos permanentes, éstos se encuentran en la librería SASUSER[27]. Para que SAS encuentre los formatos de esta librería se debe especificar la sentencia OPTIONS en el programa SAS antes de acceder a los SAS-files. Por ejemplo:

```
OPTIONS   FMTSEARCH = ( WORK    LIBRARY   SASUSER);
```

La opción FMTSEARCH especifica las librerías donde se buscan los formatos permanentes de las variables de los SAS-files.

NOTA: por defecto el valor de FMTSEARCH es (WORK LIBRARY)

[27] Por defecto, la librería SASUSER se asocia al directorio \Mis documentos\My SAS Files\V8 en la unidad donde se encuentra instalado SAS. Se puede averiguar el directorio asociado con la librería SASUSER, pulsando el botón derecho del ratón y seleccionando propiedades.